

B.M.S. COLLEGE OF ENGINEERING
Basavanagudi, Bengaluru- 560019
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING



LAB REPORT

On

Object Oriented Java Programming
(23CS3PCOOJ)

Submitted By:
AGNEYA D A
1BM22CS024

In partial fulfilment of
BACHELOR OF ENGINEERING
In
COMPUTER SCIENCE AND ENGINEERING
2023-24

Faculty-In-Charge
Swathi Sridharan

Assistant Professor
Department of Computer Science and Engineering

SL. NO	DATE	TOPIC	PageNo
1	22/12/23	Quadratic Equation	1
2	29/12/23	Student SGPA Calculator	3
3	12/01/24	Book Problem	7
4	12/01/24	Shapes	10
5	19/01/24	Bank Problem	13
6	02/02/24	Student External And Internal Marks	18
7	16/02/24	Exception Handling	24
8	16/02/24	Threads	27
9	23/02/24	AWT	30

LAB-1:QUADRATIC EQUATION

Develop a Java program that prints all real solutions to the quadratic equation $ax^2+bx+c=0$. Read in a, b, c and use the quadratic formula. If the discriminate b^2-4ac is negative, display a message stating that there are no real solutions.

```
import java.util.Scanner;

import java.lang.Math;

public class QuadEqn{

    public static void main(String[] args){

        float a, b, c, dsc, r1, r2;

        Scanner reader = new Scanner(System.in);

        System.out.println("Enter the co-efficient of x^2: ");

        a = reader.nextFloat();

        if (a==0){

            System.out.println("Invalid Input");

        }

        else{

            System.out.println("Enter the co-efficient of x: ");

            b = reader.nextFloat();

            System.out.print("Enter the value of the constant: ");

            c = reader.nextFloat();

            dsc = (float)Math.pow(b,2) - 4*a*c;

            if(dsc > 0){

                r1 = (float)(-b + Math.sqrt(dsc))/(2*a);

                r2 = (float)(-b - Math.sqrt(dsc))/(2*a);

                System.out.println("The roots are:" + r1+ " and " + r2);

            }

            else if(dsc==0){

                r1 = (float)-b/(2*a);
```

```

        System.out.println("The root is: "+ r1);
    }
    else{
        System.out.println("No real roots exist for this equation");
    }
    System.out.println("Agneya D A 1BM22CS024");
}
}
}

```

OUTPUT :

```

C:\Users\bmsce\Desktop\1BM22CS024>java QuadEqn
Enter the co-efficient of x^2:
1
Enter the co-efficient of x:
3
Enter the value of the constant:
2
The roots are r1 = -1.0 and r2 = -2.0
Agneya D A 1BM22CS024

```

```

C:\Users\bmsce\Desktop\1BM22CS024>java QuadEqn
Enter the co-efficient of x^2:
100
Enter the co-efficient of x:
1
Enter the value of the constant:
1
There are no real roots for this equation
Agneya D A 1BM22CS024

```

```

C:\Users\bmsce\Desktop\1BM22CS024>java QuadEqn
Enter the co-efficient of x^2:
1
Enter the co-efficient of x:
-2
Enter the value of the constant:
1
The root is: 1.0
Agneya D A 1BM22CS024

```

```
C:\Users\bmsce\Desktop\1BM22CS024>java QuadEqn
Enter the co-efficient of x^2:
0
Invalid input
```

LAB-2: STUDENT SGPA CALCULATION

Develop a Java program to create a class Student with members usn, name, an arraycredits and an array marks. Include methods to accept and display details and amethod to calculate SGPA of a student.

```
import java.util.Scanner;
```

```
import java.lang.Math;
```

```
class Student {
```

```
    int num_subs = 8;
```

```
    double credits[] = new double[num_subs];
```

```
    double marks[] = new double[num_subs];
```

```
    double grade[] = new double[num_subs];
```

```
    double sgpa, num = 0, den = 0;
```

```
    String name, usn;
```

```
    void accept_details() {
```

```
        Scanner reader = new Scanner(System.in);
```

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

```
        usn = reader.nextLine();
```

```
        System.out.println("Enter student name: ");
```

```
        name = reader.nextLine();
```

```
        for (int i = 0; i < num_subs; i++) {
```

```
            System.out.println("Enter number of credits: ");
```

```

        credits[i] = reader.nextDouble();

        System.out.println("Enter the marks obtained out of 100: ");

        marks[i] = reader.nextDouble();

    }
}

```

```

void display_details() {

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

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

    System.out.print("Credits: ");

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

        System.out.print(credits[i] + ", ");

    }

    System.out.println("");

    System.out.print("Marks: ");

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

        System.out.print(marks[i] + ", ");

    }

    System.out.println("");

    System.out.println("SGPA: " + calculate_sgpa());

}

```

```

double calculate_sgpa() {

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

        if (marks[i] >= 40 && marks[i] <= 100) {

            grade[i] = Math.floor(marks[i] / 10) + 1;

        } else {

            grade[i] = 0;

        }

    }

}

```

```

    }

    for (int i = 0; i < num_subs; i++) {
        num += credits[i] * grade[i];
        den += credits[i];
    }

    sgpa = num / den;
    return sgpa;
}
}

public class SgpaCalc {
    public static void main(String[] args) {
        Student agneya = new Student();
        agneya.accept_details();
        agneya.display_details();

        // System.out.println("SGPA is: "+agneya.calculate_sgpa());
    }
}

```

OUTPUT :

```
C:\Users\bmsce\Desktop\1BM22CS024\lab 2>java SgpaCalc
Enter name:
Agneya
Enter USN:
1BM22CS024
Enter Credits:
4
Enter marks:
90
Enter Credits:
4
Enter marks:
92
Enter Credits:
3
Enter marks:
87
Enter Credits:
3
Enter marks:
95
Enter Credits:
3
Enter marks:
92
Enter Credits:
1
Enter marks:
97
```

```
Enter marks:
97
Enter Credits:
1
Enter marks:
96
Enter Credits:
1
Enter marks:
95
Name: Agneya
USN: 1BM22CS024
Credits: 4.0 4.0 3.0 3.0 3.0 1.0 1.0 1.0
Marks: 90.0 92.0 87.0 95.0 92.0 97.0 96.0 95.0
SGPA: 9.85
```


LAB-3: BOOK DETAILS

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

```
import java.util.Scanner;
```

```
class Book {
```

```
    Scanner reader = new Scanner(System.in);
```

```
    String name, author;
```

```
    int price, num_pages;
```

```
    Book() {
```

```
        setDetails();
```

```
    }
```

```
    void setDetails() {
```

```
        System.out.println("Enter the name, author, price, number of pages in order: ");
```

```
        name = reader.next();
```

```
        author = reader.next();
```

```
        price = reader.nextInt();
```

```
        num_pages = reader.nextInt();
```

```
    }
```

```

public String toString() {

    return "Details of book: \nName: " + name + "\nAuthor: " + author + "\nPrice: " + price +
"\nNum pages: "

        + num_pages;

}

```

```

void getDetails() {

    System.out.println(toString());

}

}

```

```

public class BookProg {

    public static void main(String[] args) {

        int i, n;

        System.out.println("Enter the value of n: ");

        Scanner mains = new Scanner(System.in);

        n = mains.nextInt();

        Book[] books = new Book[n];

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

            Book draft = new Book();

            books[i] = draft;

        }

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

            books[i].getDetails();

        }

    }

}

```

```
    }  
  
    System.out.println("1BM22CS024 Agneya D A");  
  
    }  
  
}
```

OUTPUT :

```
Enter the value of n:  
2  
Enter the name, author, price, number of pages in order:  
James AtomicHabits 300 200  
Enter the name, author, price, number of pages in order:  
Cruyff MyTurn 400 174  
Details of book:  
Name: James  
Author: AtomicHabits  
Price: 300  
Num pages: 200  
Details of book:  
Name: Cruyff  
Author: MyTurn  
Price: 400  
Num pages: 174  
1BM22CS024 Agneya D A
```

LAB-4: AREA CALCULATION

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

```
abstract class Shapes {  
    int a, b;  
  
    void printArea() {  
    }  
}  
  
class Triangle extends Shapes {  
    int a, b;  
  
    Triangle(int base, int alt) {  
        a = base;  
        b = alt;  
    }  
  
    void printArea() {  
        System.out.println("Area =" + (0.5 * a * b));  
    }  
}  
  
class Rectangle extends Shapes {  
    int a, b;
```

```

Rectangle(int height, int width) {
    a = height;
    b = width;
}

void printArea() {
    System.out.println("Area =" + (a * b));
}
}

class Circle extends Shapes {
    int a, b;

    Circle(int radius) {
        a = radius;
        b = 0;
    }

    void printArea() {
        System.out.println("Area =" + (3.14 * a * a));
    }
}

public class ShapeProg {
    public static void main(String[] args) {
        Triangle t = new Triangle(5, 5);
        Rectangle r = new Rectangle(5, 5);
        Circle c = new Circle(5);
    }
}

```

```
t.printArea();  
r.printArea();  
c.printArea();  
System.out.println("1BM22CS024 Agneya D A");  
}  
}
```

OUTPUT :

```
Area =12.5  
Area =25  
Area =78.5  
1BM22CS024 Agneya D A
```

LAB-5: BANK ACCOUNT DETAILS

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

- a) Accept deposit from customer and update the balance.
- b) Display the balance.
- c) Compute and deposit interest
- d) Permit withdrawal and update the balance

Check for the minimum balance, impose penalty if necessary and update the balance.

```
import java.util.Scanner;
```

```
import java.lang.Math;
```

```
class Account {  
    String name;  
    long accno;  
    String acctype;  
    double bal;  
  
}
```

```
class CurrAcct extends Account {
```

```

double min_balance = 1000;
double charge_rate = 0.05;

CurrAcct(String nam, long accnum, double start) {
    acctype = "current";
    bal = start;
    name = nam;
    accno = accnum;
}

public void withdraw(double amount) {
    bal -= amount;
    if (bal < min_balance) {
        charge();
    }
}

public void charge() {
    bal = bal - charge_rate * bal;
}

boolean cheque_book = true;

void deposit(double amount) {
    bal += amount;
}
}

class SavAcct extends Account {

```



```
double interest_rate = 0.04;
```

```
public SavAcct(String nam, long accnum, double start) {
```

```
    name = nam;
```

```
    accno = accnum;
```

```
    bal = start;
```

```
    acctype = "savings";
```

```
}
```

```
public void addInterest(double time) {
```

```
    bal = bal * Math.pow((1 + interest_rate / 4), 4 * time);
```

```
}
```

```
boolean cheque_book = false;
```

```
void deposit(double amount) {
```

```
    bal += amount;
```

```
}
```

```
public void withdraw(double amount) {
```

```
    bal -= amount;
```

```
}
```

```
}
```

```
public class Bank {
```

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

```
        Scanner reader = new Scanner(System.in);
```

```
        System.out.println("opening a savings account");
```

```
        System.out.println("Enter your name: ");
```

```

String name = reader.nextLine();
System.out.println("Account number: ");
long accnum = reader.nextLong();
System.out.println("initial Deposit: ");
double start = reader.nextDouble();
SavAcct sav = new SavAcct(name, accnum, start);
System.out.println("Opening a current account");
System.out.println("Enter your name: ");
name = reader.nextLine();
System.out.println("Account number: ");
accnum = reader.nextLong();
System.out.println("initial Deposit: ");
start = reader.nextDouble();
CurrAcct cur = new CurrAcct(name, accnum, start);
System.out.println("What amount should be deposited into the savings account? ");
double amount = reader.nextDouble();
sav.deposit(amount);
System.out.println("What amount should be deposited to the current account? ");
amount = reader.nextDouble();
cur.deposit(amount);
System.out.println("Withdrawing 2000 from savings account");
sav.withdraw(2000);
System.out.println("Withdrawing " + (amount + 2001) + " from current account");
cur.withdraw(amount + 2001);
System.out.println("How many years have passed since depositing in the savings
account? ");
double years = reader.nextDouble();
sav.addInterest(years);
System.out.println("Amount in savings account: " + sav.bal);
System.out.println("Amount in current account: " + cur.bal);

```

```
}  
}
```

OUTPUT :

```
PS C:\Users\bmsce\Desktop\1BM22CS024\lab 4> cd "c:\Users\bmsce\Desktop\1BM22CS024\lab 4"
Opening a savings account!
Enter your name:
Agneya
Account number:
1
Initial deposit:
3000
Opening a current account!
Account number:
2
Initial deposit:
3000
What amount should be deposited to the Savings account?
400
What amount should be deposited to the current account?
500
Withdrawing 2000 from sav
Withdrawing 2501.0 from cur
Enter the no. of years since depositing in sav:
7
Amount in savings account = 1849.8073536575516
Amount in current account = 949.05
1BM22CS024 Agneya D A
PS C:\Users\bmsce\Desktop\1BM22CS024\lab 4> |
```

LAB-6: CALCULATION OF MARKS

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.

```
//cie PACKAGE
```

```
// Student.java
```

```
package cie;
```

```
public class Student {
```

```
    public String name;
```

```
    public String usn;
```

```
    public int sem;
```

```
    public void setStudent(String nam, String sn, int semester) {
```

```
        name = nam;
```

```
        usn = sn;
```

```
        sem = semester;
```

```
    }
```

```
}
```

```
//Internals.java
```

```
package cie;
```

```
import java.util.Scanner;
```

```
public class Internals extends Student{
```

```
    public Scanner reader = new Scanner(System.in);
```

```

public int[] inmarks = new int[5];
public void setInternals(){
    for(int i=0; i<5; i++){
        System.out.println("Enter internal marks of course " + (i+1) + ":");
        inmarks[i] = reader.nextInt();
    }
}
}

```

```

//see PACKAGE
//Externals.java
package see;
import java.util.Scanner;
import cie.*;
public class Externals extends Student{
    public Scanner reader = new Scanner(System.in);
    public int[] exmarks = new int[5];
    public void setExternals(){
        for(int i=0; i<5; i++){
            System.out.println("Enter external marks of course " + (i+1) + ":");
            exmarks[i] = reader.nextInt();
        }
    }
}

```

```

//Marks.java
import cie.*;
import see.*;
import java.util.Scanner;

```

```

import java.lang.Math;

public class Marks {

    public static void main(String[] args) {

        int i, j;

        Scanner reader = new Scanner(System.in);

        System.out.println("Enter the value of n");

        int n = reader.nextInt();

        Internals[] intarr = new Internals[n];

        Externals[] extarr = new Externals[n];

        String name, usn;

        int semester;

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

            System.out.println("Enter name: ");

            name = reader.nextLine();

            name = reader.nextLine();

            System.out.println("Enter usn: ");

            usn = reader.nextLine();

            // usn = reader.nextLine();

            System.out.println("Enter semester: ");

            semester = reader.nextInt();

            Internals studin = new Internals();

            studin.setInternals();

            Externals studex = new Externals();

            studex.setExternals();

            studin.setStudent(name, usn, semester);

            studex.setStudent(name, usn, semester);

            intarr[i] = studin;
        }
    }
}

```

```
        extarr[i] = studex;
    }

    for (i = 0; i < n; i++) {
        System.out.println("Name: " + intarr[i].name);
        System.out.println("USN: " + intarr[i].usn);
        System.out.println("Sem " + intarr[i].sem);

        for (j = 0; j < 5; j++) {
            System.out.println("Course " + (j + 1) + ":"
                + (intarr[i].inmarks[j] + Math.ceil(((double) extarr[i].exmarks[j] / 2))));
        }
        System.out.println("");
    }

    System.out.println("1BM22CS024 Agneya D A");
}
}
```

OUTPUT :

```
Enter the value of n
2
Enter name:
Agneya
Enter usn:
024
Enter semester:
3
Enter internal marks of course 1:
45
Enter internal marks of course 2:
46
Enter internal marks of course 3:
47
Enter internal marks of course 4:
48
Enter internal marks of course 5:
49
Enter external marks of course 1:
100
Enter external marks of course 2:
99
Enter external marks of course 3:
98
Enter external marks of course 4:
97
Enter external marks of course 5:
96
Enter name:
Aman
Enter usn:
034
Enter semester:
1
Enter internal marks of course 1:
45
```



```
Enter internal marks of course 2:
46
Enter internal marks of course 3:
47
Enter internal marks of course 4:
48
Enter internal marks of course 5:
49
Enter external marks of course 1:
70
Enter external marks of course 2:
75
Enter external marks of course 3:
80
Enter external marks of course 4:
85
Enter external marks of course 5:
90
Name: Agneya
USN: 024
Sem 3
Course 1:95.0
Course 2:96.0
Course 3:96.0
Course 4:97.0
Course 5:97.0
```

```
Name: Aman
USN: 034
Sem 1
Course 1:80.0
Course 2:84.0
Course 3:87.0
Course 4:91.0
Course 5:94.0
```

```
1BM22CS024 Agneya D A
PS C:\Users\bmsce\Desktop\1BM22CS024\lab 6> █
```

LAB-7: EXCEPTION HANDLING

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 takes both father and son’s age and throws an exception if son’s age is >=father’s age.

```
import java.util.Scanner;
```

```
class WrongAge extends Exception {  
    WrongAge(String errorMessage) {  
        super(errorMessage);  
    }  
}
```

```
class ParentAge extends Exception {  
    ParentAge(String errorMessage) {  
        super(errorMessage);  
    }  
}
```

```
class Father {  
    int age;  
  
    Father(int num) throws WrongAge {  
        if (num < 0) {  
            throw new WrongAge("Age can't be less than zero");  
        } else {  
            age = num;  
        }  
    }  
}
```

```

    }
}

int fatherAge() {
    return age;
}
}

class Son extends Father {

    int age;

    Son(int num1, int num2) throws ParentAge, WrongAge {
        super(num1);
        if (num2 > super.age) {
            throw new ParentAge("The son's age cannot be greater than the father's.");
        } else {
            this.age = num2;
        }
    }
}

public class Exception1 {

    public static void main(String[] args) throws WrongAge, ParentAge {
        Son s1 = null;
        Son s2 = null;
        Son s3 = null;
        progBody("Case 1: ", s1, -10, 30);
        progBody("Case 2: ", s2, 10, 20);
        progBody("Case 3: ", s3, 45, 19);
    }
}

```

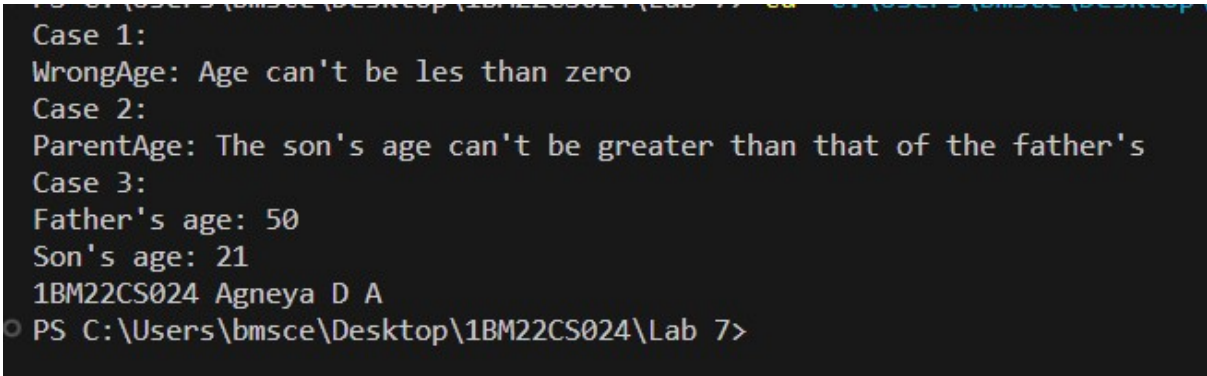
```

    }

    static void progBody(String disp, Son s1, int f_age, int s_age) {
        System.out.println(disp);
        try {
            s1 = new Son(f_age, s_age);
        } catch (WrongAge errorText) {
            System.out.println(errorText);
        } catch (ParentAge errortext) {
            System.out.println(errortext);
        } finally {
            if (s1 != null) {
                System.out.println("Father's age: " + s1.fatherAge());
                System.out.println("Son's age: " + s1.age);
            }
        }
    }
}

```

OUTPUT :



```

PS C:\Users\bmsce\Desktop\1BM22CS024\Lab 7> java -cp . Son
Case 1:
WrongAge: Age can't be les than zero
Case 2:
ParentAge: The son's age can't be greater than that of the father's
Case 3:
Father's age: 50
Son's age: 21
1BM22CS024 Agneya D A
PS C:\Users\bmsce\Desktop\1BM22CS024\Lab 7>

```

LAB-8: MULTITHREADING

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 Bmsc implements Runnable {  
    String name;  
    Thread t;  
  
    Bmsc(String threadName) {  
        this.name = threadName;  
        t = new Thread(this, this.name);  
        t.start();  
    }  
  
    synchronized public void run() {  
        while (true) {  
            System.out.println("BMS College of Engineering");  
            try {  
                Thread.sleep(10000);  
            } catch (InterruptedException e) {  
                e.printStackTrace();  
            }  
        }  
    }  
}  
  
class Cse implements Runnable {
```

```

String name;

Thread t;

Cse(String threadName) {
    this.name = threadName;
    t = new Thread(this, this.name);
    t.start();
}

synchronized public void run() {
    while (true) {
        System.out.println("CSE");
        try {
            Thread.sleep(2000);
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
    }
}

}

public class Thread1 {
    public static void main(String[] args) {
        Bmsc bms = new Bmsc("BmsC");
        Cse cse = new Cse("CompSc");
    }
}

```

OUTPUT :

```
1BM22CS024 Agneya D A
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
}
```

LAB-9: AWT

09) 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 javax.swing.*;
import java.awt.*;
import java.awt.event.*;

class SwingDemo{
    SwingDemo(){
        // create jframe container
        JFrame jfrm = new JFrame(""Divider App"");
        jfrm.setSize(275, 150);
        jfrm.setLayout(new FlowLayout());
        // to terminate on close
        jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        // text label
        JLabel jlab = new JLabel(""Enter the divider and dividend:"");

        // add text field for both numbers
        JTextField ajtf = new JTextField(8);
        JTextField bjtf = new JTextField(8);
```



```

// calc button
JButton button = new JButton("&quot;Calculate&quot;");

// labels
JLabel err = new JLabel();
JLabel alab = new JLabel();
JLabel blab = new JLabel();

JLabel anslab = new JLabel();

// add in order :)
jfrm.add(err); // to display error bois
jfrm.add(jlab);
jfrm.add(ajtf);
jfrm.add(bjtf);
jfrm.add(button);
jfrm.add(alab);
jfrm.add(blab);
jfrm.add(anslab);

ActionListener l = new ActionListener() {
    public void actionPerformed(ActionEvent evt) {
        System.out.println("&quot;Action event from a text field&quot;");
    }
};
ajtf.addActionListener(l);
bjtf.addActionListener(l);

```

```

button.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent evt) {
        try{
            int a = Integer.parseInt(ajtf.getText());
            int b = Integer.parseInt(bjtf.getText());
            int ans = a/b;

            alab.setText("&quot;\nA = &quot; + a);
            blab.setText("&quot;\nB = &quot; + b);
            anslab.setText("&quot;\nAns = &quot;+ ans);
        }
        catch(NumberFormatException e){
            alab.setText("&quot;&quot;);
            blab.setText("&quot;&quot;);
            anslab.setText("&quot;&quot;);

            err.setText("&quot;Enter Only Integers!&quot;);
        }
        catch(ArithmeticException e){
            alab.setText("&quot;&quot;);
            blab.setText("&quot;&quot;);
            anslab.setText("&quot;&quot;);
            err.setText("&quot;B should be NON zero!&quot;);
        }
    }
});

// display frame
jfrm.setVisible(true);

```

```
}

public static void main(String args[]){
    // create frame on event dispatching thread
    SwingUtilities.invokeLater(new Runnable(){
        public void run(){
            new SwingDemo();
        }
    });
}
}
```

