

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:

AAKANKSHA V R
1BM22CS001

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 | 19 |
| 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;

        System.out.println("Aakanksha V R 1BM22CS001");

        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");
    }
}
}
}

```

OUTPUT :

```

C:\Users\bmsce\Desktop\1BM22CS001>java QE
Enter a,b,c
2
4
2
Aakanksha V R 1BM22CS001
Roots are real and equal
Roots are-4.0 & -4.0

```

```

C:\Users\bmsce\Desktop\1BM22CS001>java QE
Enter a,b,c
2
7
1
Aakanksha V R 1BM22CS001
Roots are real and distinct
Roots are-0.5968758 & -13.403124

```

```

C:\Users\bmsce\Desktop\1BM22CS001>java QE
Enter a,b,c
1
2
3
Aakanksha V R 1BM22CS001
Roots are imaginary
Roots are-1.0+i1.4142135 & -1.0-i1.4142135

```

```
C:\Users\bmsce\Desktop\1BM22CS001>java QE
Enter a,b,c
0
2
3
Aakanksha V R 1BM22CS001
Invalid input for a
```

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 S1= new Student();
        S1.accept_details();
        S1.display_details();
        // System.out.println("SGPA is: "+S1.calculate_sgpa());
    }
}

```

OUTPUT :

```
Enter USN:
CS001
Enter Name:
Aakanksha.V.R
Enter details:

Enter the credits of subject1:
4
Enter the marks of subject1:
95
Enter the credits of subject2:
4
Enter the marks of subject2:
97
Enter the credits of subject3:
3
Enter the marks of subject3:
90
Enter the credits of subject4:
3
Enter the marks of subject4:
85
Enter the credits of subject5:
3
Enter the marks of subject5:
84
Enter the credits of subject6:
```



```
Enter the credits of subject3:
3
Enter the marks of subject3:
90
Enter the credits of subject4:
3
Enter the marks of subject4:
85
Enter the credits of subject5:
3
Enter the marks of subject5:
84
Enter the credits of subject6:
1
Enter the marks of subject6:
99
Enter the credits of subject7:
1
Enter the marks of subject7:
95
Enter the credits of subject8:
1
Enter the marks of subject8:
90
Student details:
USN: CS001
nAME: Aakanksha.V.R
SGPA: 9.7
```

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.*;
```

```

class Book01 {
String name;
String author;
    float price;
    int num_pages;
    void set_details({}

Scanner sc = new Scanner(System.in); // Added System.in to Scanner constructor

    System.out.println("Enter bookname, author, price, num_pages");
    name = sc.next();
    author = sc.next();
    price = sc.nextFloat();
    num_pages = sc.nextInt();
}

void get_details() {
    String details = toString();
    System.out.println(details);
}

public String toString() {
    return "The book " + name + " was written by " + author + ". It consists of " +
num_pages + " pages and costs around $" + price;
}

public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    System.out.println("Enter number of books");
    int n = sc.nextInt();
    Book01 b[] = new Book01[n];
    for (int i = 0; i < n; i++) {
        b[i] = new Book01();
        b[i].set_details();
    }
}

```

```

        System.out.println("Book details");

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

            b[i].get_details();

        }

        System.out.println("Aakanksha V R,1BM22CS001");

    }

}

```

OUTPUT :

```

C:\Users\skc\Desktop\ooj p>java Book01
Enter number of books
2
Enter bookname, author, price, num_pages
ABC DEF 250 100
Enter bookname, author, price, num_pages
XYZ VWE 355 200
Book details
The book ABC was written by DEF. It consists of 100 pages and costs around $250.0
The book XYZ was written by VWE. It consists of 200 pages and costs around $355.0
Aakanksha V R,1BM22CS001

C:\Users\skc\Desktop\ooj p>

```

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.

```
import java.util.Scanner;

abstract class Shape
{
    int x,y;
    abstract void PrintArea();
    public static void main(String args[])
    {
        System.out.println("Aakanksha V R,1BM22CS001");
        Shape obj1 = new Circle();
        obj1.PrintArea();
        Shape obj2 = new Rectangle();
        obj2.PrintArea();
        Shape obj3= new Triangle();
        obj3.PrintArea();

    }

}

class Circle extends Shape{
    Circle()
    {
        Scanner sc = new Scanner(System.in);
```

```

System.out.println("enter the radius of the circle");
x=sc.nextInt();
y=x;
}

void PrintArea()
{System.out.println("area of the circle is"+ 3.14*x*y);
}
}

class Rectangle extends Shape{
Rectangle()
{
Scanner sc = new Scanner(System.in);
System.out.println("enter the length and breadth of Rectangle");
x=sc.nextInt();
y=sc.nextInt();
}
void PrintArea()
{
System.out.println("area of the rectangle is"+ x*y);
}
}

class Triangle extends Shape{
Triangle()
{
Scanner sc = new Scanner(System.in);
System.out.println("enter the base and height of Triangle");

x=sc.nextInt();

```

```
y=sc.nextInt();  
}  
void PrintArea()  
{  
System.out.println("area of the triangle is"+ 0.5*x*y);  
}}
```

```
C:\Users\skc\Desktop\ooj p>java Shape  
Aakanksha V R,1BM22CS001  
enter the radius of the circle  
5  
area of the circle is78.5  
enter the length and breadth of Rectangle  
6  
7  
area of the rectangle is42  
enter the base and height of Triangle  
9  
8  
area of the triangle is36.0
```

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;

class Account
{
    String customerName;
    long accno;
    String accountType;
    double balance;

    public Account (String customerName, long accno, String accountType)
    {
        this.customerName = customerName;
        this.accno = accno;

        this.accountType = accountType;
        this.balance = 0.0;
```

```

    }

    public void displayBalance ()
    {
        System.out.println ("Account Number: " + accno);
        System.out.println ("Customer Name: " + customerName);
        System.out.println ("Account Type: " + accountType);
        System.out.println ("Balance: $" + balance);
    }
}

class CurAcct extends Account
{
    double minBalance;
    double serviceCharge;
    public CurAcct (String customerName, long accno)
    {
        super (customerName, accno, "Current");
        this.minBalance = 500.0
        this.serviceCharge = 50.0
    }
    public void withdraw (double amount)
    {
        if (balance - amount >= minBalance)
        {
            balance -= amount;
            System.out.println ("Withdrawal successful. Current Balance: $" +balance);
        }
        else
        {
            System.out.println ("Insufficient funds. Withdrawal not allowed.");
        }
    }
}

```



```

    }
}

public void imposeServiceCharge ()
{
    if (balance < minBalance)
    {
        balance -= serviceCharge;
        System.out.println ("Service charge imposed. Current Balance: Rs." +balance);
    }
}

class SavAcct extends Account
{
    double interestRate;

    public SavAcct (String customerName, long accno)
    {
        super (customerName, accno, "Savings");
        this.interestRate = 0.05;
    }

    public void depositInterest ()
    {
        double interest = balance * interestRate;
        balance += interest;
        System.out.println ("Interest deposited. Current Balance: $" + balance);
    }

    public void compoundInterest (double initialAmount, int term)
    {
        double compoundInterest =initialAmount * Math.pow ((1 + interestRate), term) -
        initialAmount;

        balance += compoundInterest;
    }
}

```

```

System.out.println ("Compound Interest deposited. Current Balance: Rs." +balance);
}
}

public class Bank01
{
public static void main (String[]args)
{
Scanner scanner = new Scanner (System.in);
System.out.println("Aakanksha V R,1BM22CS001");
System.out.println ("Choose account type:");
System.out.println ("1. Current");
System.out.println ("2. Savings");

System.out.print ("Enter choice (1 or 2): ");
int choice = scanner.nextInt ();
System.out.print ("Enter customer name: ");
String customerName = scanner.next ();
System.out.print ("Enter account number: ");
long accno = scanner.nextLong ();
if (choice == 1)
{
CurAcct curAccount = new CurAcct (customerName, accno);
System.out.print ("Enter initial balance: $");
double initialBalance = scanner.nextDouble ();
curAccount.balance = initialBalance;
System.out.print ("Enter withdrawal amount: $");
double withdrawalAmount = scanner.nextDouble ();
curAccount.withdraw (withdrawalAmount);
curAccount.imposeServiceCharge ();
}
}
}

```

```

curAccount.displayBalance ();
}
else if (choice == 2)
{
SavAcct savAccount = new SavAcct (customerName, accno);
System.out.print ("Enter initial balance: $");
double initialBalance = scanner.nextDouble ();
savAccount.balance = initialBalance;
System.out.print ("Enter withdrawal amount: $");
double withdrawalAmount = scanner.nextDouble ();
savAccount.balance -= withdrawalAmount;
System.out.println ("Withdrawal successful. Current Balance: $" +savAccount.balance);
System.out.print ("Enter interest rate: ");
double interestRate = scanner.nextDouble ();
savAccount.interestRate = interestRate;
savAccount.displayBalance ();
System.out.print ("Enter term (in years) for compound interest calculation: ");
int term = scanner.nextInt ();
savAccount.compoundInterest (initialBalance, term);
savAccount.displayBalance ();
}
else
{
System.out.println ("Invalid choice");
}
}

```

OUTPUT :

```
C:\Users\skc\Desktop\ooj p>java Bank01
Aakanksha V R,1BM22CS001
Choose account type:
1. Current
2. Savings
Enter choice (1 or 2): 2
Enter customer name: CD
Enter account number: 6
Enter initial balance: $1500
Enter withdrawal amount: $1100
Withdrawal successful. Current Balance: $400.0
Enter interest rate: 5
Account Number: 6
Customer Name: CD
Account Type: Savings
Balance: $400.0
Enter term (in years) for compound interest calculation: 2
Compound Interest deposited. Current Balance: Rs.52900.0
Account Number: 6
Customer Name: CD
Account Type: Savings
Balance: $52900.0
```

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.

```
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;
        System.out.println("Aakanksha V R");
        System.out.println("1Bm22cs001");
        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("");
    }

}
}

```

OUTPUT :


```
Aakanksha
1bm22cs001
enter no of students:
1
Enter details1
Enter sem,usn and name:

3
1bm22cs001
Enter internal and see marks of sub1
45
45
Enter internal and see marks of sub2
46
46
Enter internal and see marks of sub3
47
47
Enter internal and see marks of sub4
48
48
Enter internal and see marks of sub5
49
49
Final marks of 1bm22cs001
Course1=90
Course2=92
Course3=94
Course4=96
Course5=98
```

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 cases 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 :

```

C:\Users\bmsce\Desktop\1BM22CS001>java fatherson
Enter father's age and son's age
45
32
Father's age:45
Son's age :32
Aakanksha V R,1BM22CS001

C:\Users\bmsce\Desktop\1BM22CS001>java fatherson
Enter father's age and son's age
-8
22
ERROR:Age cannot be negative
Aakanksha V R,1BM22CS001

C:\Users\bmsce\Desktop\1BM22CS001>java fatherson
Enter father's age and son's age
23
25
ERROR:Son's age must be less than father's age
Aakanksha V R,1BM22CS001

```

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 A extends Thread
```

```
{
```

```
    int t1,time;
```

```
    A(){
```

```
        t1=0;
```

```
        time=25000;
```

```
    }
```

```
    public void run()
```

```
{
```

```

        while(t1<=time)
        {
            System.out.println("BMS COLLEGE OF ENGINEERING");
            try {
                sleep(10000);
            } catch(Exception e) {
                System.out.println("error");
            }
            t1+=10000;
        }
    }
}

class B extends Thread{
    int t2,time;

    B(){
        time=25000;
        t2=0;
    }

    public void run()
    {
        while(t2<=time)

        {
            System.out.println("CSE");
            try{
                sleep(2000);
            }
            catch(Exception e)
            {
                System.out.println("error");
            }
        }
    }
}

```

```

        }
        t2+=2000;
    }}
}
class program8demo
{
    public static void main(String args[])
    {
        A a=new A();
        B b=new B();
        a.start();
        b.start();
        System.out.println("Aakanksha V R,1BM22CS001");
    }
}

```

OUTPUT :

```

C:\Users\bmsce\Desktop\1BM22CS001>java program8demo
Aakanksha V R,1BM22CS001
CSE
BMS COLLEGE OF ENGINEERING
CSE
CSE
CSE
CSE
BMS COLLEGE OF ENGINEERING
CSE
CSE
CSE
CSE
CSE
BMS COLLEGE OF ENGINEERING
CSE
CSE
CSE
}

```

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 Arithmetic Exception Display the exception in a message dialog box.

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

class SwingDemo{
    SwingDemo(){
        JFrame jfrm=new JFrame("Divider App");
        jfrm.setSize(275,150);
        jfrm.setLayout(new FlowLayout());
        jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        JLabel jlab=new JLabel("Enter the dividend and divisor:");
        JLabel plab=new JLabel("Aakanksha V R,1BM22CS001");

        JTextField ajtf=new JTextField(8);
        JTextField bjtf=new JTextField(8);

        JButton button=new JButton("Calculate");

        JLabel err=new JLabel();
        JLabel alab=new JLabel();
        JLabel blab=new JLabel();
```



```

JLabel anslab=new JLabel();

jfrm.add(err);
jfrm.add(jlab);
jfrm.add(plab);
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("Action event from atext field");}

};

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("\n A="+a);
            blab.setText("\n B="+b);
            anslab.setText("\n Ans="+ans);}

```

```

        catch(NumberFormatException e){
            alab.setText("");
            blab.setText("");
            anslab.setText("");
            err.setText("Enter only Integers!");
        }

        catch(ArithmeticException e){
            alab.setText("");
            blab.setText("");
            anslab.setText("");
            err.setText("B should ne NON zero!");}
    }

});

jfrm.setVisible(true);
}

public static void main(String args[]){
    SwingUtilities.invokeLater(new Runnable(){
        public void run(){
            new SwingDemo();
            System.out.println("Aakanksha V R ,1BM22CS001");
        }
    });
}
}

```

B should ne NON zero! Enter the dividend and divisor: Aakanksha V R,1BM22CS001

Enter only Integers! Enter the dividend and divisor: Aakanksha V R,1BM22CS001 A=24 B=5 Ans=4

Enter only Integers! Enter the dividend and divisor: Aakanksha V R,1BM22CS001