

**B.M.S COLLEGE OF ENGINEERING BENGALURU**  
Autonomous Institute, Affiliated to VTU



# **JAVA LAB REPORT**

## **OBJECT ORIENTED JAVA PROGRAMMING**

Bachelor of Engineering  
in  
Computer Science and Engineering

*Submitted by:*

**Rishi J**  
**1BM22CS222**

*Submitted to:*

**Dr. Seema Patil**  
Assistant Professor  
BMS College of Engineering

## **INDEX**

<b>Sl.No.</b>	<b>Title</b>	<b>Date</b>
1	Complete scanned Observation Book	12/12/2023 - 20/02/2024
2	Lab 1	12/12/2023
3	Lab 2	19/12/2023
4	Lab 3	26/12/2023
5	Lab 4	02/01/2024
6	Lab 5	09/01/2024
7	Lab 6	16/01/2024
8	Lab 7	23/01/2024
9	Lab 8	30/01/2024
10	Lab 9	06/02/2024
11	Lab 10	20/02/2024

Java - Date - 12/12/23

; (0+5)(0+6b) / (([b] input. num) - (d-)) = 50  
; ("Enter the two real nos. Root") attrif. to . method  
; (d+5) + 5 \* 6b + 5 \* 5 = 100d " attrif. to . method  
; (0+5) \ (d-) = 100  
) Develop a Java Program that prints all real solutions to the quadratic  
equation  $a^2 + bx + c = 0$ . Read a, b, c.  
(a>b) if else

import java.util.\*;

public class Quadratic

{

public static void main (String args [])

Scanner sc = new Scanner (System.in);

int a, b, c;

double r1, r2, d;

void getd()

{

System.out.println (" Enter coefficients of a,b,c ");

a = sc.nextInt();

b = sc.nextInt();

c = sc.nextInt();

}

void compute ()

{

while (a == 0)

{

System.out.println (" Not a quadratic equation ");

System.out.println (" Enter a non zero value for a ");

a = sc.nextInt();

d = (b\*b) - (4\*a\*c);

if (d == 0)

{

r1 = (-b) / (2\*a);

System.out.println (" Roots are real and equal ");

System.out.println (" Root1 = Root2 = "+r1);

else if (d > 0)

{

r1 = (-b) + (Math.sqrt(d)) / (double) (2\*a);

```

    r1 = ((-b) - (Math.sqrt(d))) / (double)(2*a);           ES1, SI, SI - std
System.out.println("Roots are real and distinct");
System.out.println("Root1 = " + r1 + " Root2 = " + r2);
}
else if (d < 0)
{
    System.out.println("Roots are imaginary");
    r1 = (-b) / (2*a);
    r2 = Math.sqrt(-d) / (2*a);
    System.out.println("Root1 = " + r1 + " + i " + r2 + " Root2 = " + r1 + " - i " + r2);
}

```

Output - Enter coefficients of  $a, b, c$

4 5 ; 6 (c, d, p) are coefficients of "1" entered. Two roots are imaginary  
 Roots are imaginary  
 Root 1 =  $0.0 + i1.053$       Root 2 =  $0.0 - i1.053$

Enter coefficients of  $a, b, c$

1 -2 1

Roots are real and equal  
 Root 1 = Root 2 = 1.0

~~Enter coefficients of a,b,c  
1 -3 2  
Root "are real and distinct" allow). two - real  
Root 1 = 2.0 Root2 = 1.0~~

$$\begin{aligned} & \left( (20 * 10) + 10 \right) - (4 * 2) = b \\ & (200 + 10) - 8 = b \\ & 210 - 8 = b \\ & b = 202 \end{aligned}$$

; ("large and here we stand") arrived. Two months later; ("large" = "stand" = "here we stand") arrived. Two months

(e.g. b)  $\mu$  m/s

$(G \times \{1\}) \cap \text{slab}(k) = \{((1), k) \in G \times \text{slab}(k) : (1, k) \in \text{slab}(k)\} = \{1\}$

## Lab Program - 2

### SGPA Calculator

```

import java.util.Scanner;
class Student {
    String usn, name;
    int n;
    Student () {
        String subject [] = new String [10];
        int credits [] = new int [10];
        int marks [] = new int [10];
    }
    void read () {
        Scanner input = new Scanner (System.in);
        System.out.println ("Enter your name : ");
        name = input.nextLine ();
        System.out.println ("Enter your USN ");
        usn = input.nextLine ();
        System.out.println ("Enter number of subjects ");
        n = input.nextInt ();
        input.nextLine ();
        for (int i = 0; i < n; i++) {
            System.out.print ("Enter subject : ");
            subject [i] = input.nextLine ();
            System.out.print ("Enter credits : ");
            credits [i] = input.nextInt ();
            System.out.print ("Enter marks : ");
            marks [i] = input.nextInt ();
            input.nextLine ();
        }
    }
}

```

number digit 2 a digit  
1111  
2 : digit > num  
Zero & 0 digit  
battent digit

```

void display () {
    System.out.println ("Name : " + name); // (.) prints name to standard output
    System.out.println ("USN : " + usn + "\n"); // prints USN to standard output
    System.out.println ("Subject |t Credits |t Marks"); // prints header to standard output
    for (int i = 0; i < n; i++) {
        System.out.println (subject [i] + " |t " + credits [i] + " |t " + marks (i));
    }
}

int grade (int a) {
    while (a/10 != 0) {
        a = a/10;
    }
    if ((a+1) > 5) {
        return (a+1);
    }
    else if ((a+1) == 5) {
        return a;
    }
    else {
        return 0;
    }
}

void calculate () {
    int total_credits = 0;
    int marks_credits = 0;
    for (int i = 0; i < n; i++) {
        total_credits += credits [i];
        marks_credits += grade (marks [i]) * credits [i];
    }
    System.out.println ("SGPA is " + (double) marks_credits / total_credits);
}

```

class marks {

```
public static void main (String [] args){  
    Scanner sc = new Scanner (System.in);  
    System.out.println ("Enter name : ");  
    String name = sc.nextLine();  
    System.out.println ("Enter USN : ");  
    String USN = sc.nextLine();  
    System.out.println ("Enter number of subjects : ");  
    int n = sc.nextInt();  
    System.out.println ("Enter subjects : ");  
    String subject [ ] = new String [n];  
    for (int i = 0; i < n; i++) {  
        System.out.println ("Subject " + (i+1) + " : ");  
        subject [i] = sc.nextLine();  
    }  
    System.out.println ("Enter credits : ");  
    int credits [ ] = new int [n];  
    for (int i = 0; i < n; i++) {  
        credits [i] = sc.nextInt();  
    }  
    System.out.println ("Enter marks : ");  
    int marks [ ] = new int [n];  
    for (int i = 0; i < n; i++) {  
        marks [i] = sc.nextInt();  
    }  
    int sum = 0;  
    for (int i = 0; i < n; i++) {  
        sum = sum + (marks [i] * credits [i]);  
    }  
    float SGPA = (float) sum / (float) n;  
    System.out.println ("SGPA is " + SGPA);  
}
```

Output -

Enter your name: Rishi

{ (0 tri) sharp }

Enter your USN: 1BM22CS222

{ (0 = 1010) stick }

Enter number of subjects : 3

: 010 = 0

Enter subject : Math

{ (2 < (1+0)) } j

Enter credits : 4

: (1+0) venture

Enter marks : 80

{ (2 = - (1+0)) } j

Enter subject: Chemistry

: 0 marks

Enter credits : 4

{ : 0 marks }

Enter marks : 95

{ : 0 marks } j

Enter subject: Physics

{ () stickles }

Enter credits : 4

; 0 = stickles - total 10

Enter marks : 85

; 0 = stickles - stickles 10

(++i; i > i; i + 1) 10

Name : Rishi

USN : 1BM22CS222

Subjects	Credits	Marks
Math	4	80
Physics	4	95
Chemistry	4	85
Total	12	260

SGPA is 9.15

SGPA is 9.15  
19.12 " is 9.12 " ) output. true. note

```

Lab Program -3

import java.util.Scanner();
public class Book {
    String name, author;
    int price, numPages;
    Book (String name, String author, int price, int numPages) {
        this.name = name;
        this.author = author;
        this.price = price;
        this.numPages = numPages;
    }
    public String toString() {
        System.out.println("Name : " + this.name);
        System.out.println("Author : " + this.author);
        System.out.println("Price : " + this.price);
        System.out.println("Pages : " + this.numPages);
        return this.name + this.author + this.price + this.numPages;
    }
}

class BookMain {
    public static void main (String args[]) {
        System.out.println("Enter no. of book objects : ");
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        sc.nextLine();
        for (int i=0; i<n; i++) {
            System.out.println("Enter book details : ");
            String name = sc.nextLine();
            String author = sc.nextLine();
            int price = sc.nextInt();
            int numPages = sc.nextInt();
            Book book = new Book(name, author, price, numPages);
            System.out.println(book);
        }
    }
}

```

String name, author;  
 int price, numPages;  
 System.out.println ("Enter name of the book");  
 name = sc.nextLine();  
 System.out.println ("Enter name of the author");  
 author = sc.nextLine();  
 System.out.println ("Enter the price");  
 price = sc.nextInt();  
 System.out.println ("Enter the number of pages");  
 numPages = sc.nextInt();  
 books[i] = new Book(name, author, price, numPages);  
}

for (i=0; i<n; i++)  
 {  
 System.out.println ("Book " + (i+1) + " Details: ");  
 books[i].toString();  
 System.out.println ();  
 }
}

Output - Enter number of books  
 Rishi - IBM22CS222  
 Enter no. of books  
 2  
 Enter name of the book  
 a1  
 Enter name of the author  
 b1  
 Enter the price  
 100  
 Enter the number of pages  
 500

Enter name of the book  
a2

Enter name of the author  
b2

Enter the price  
200

Enter the number of pages  
300

Book 1 details :

Name : a1

Author : b1

Price : 100

Pages : 500

Book 2 details :

Name : a2

Author : b2

Price : 200

Pages : 300

15-12-2021

## LAB - 4

```
import java.util.Scanner;  
class InputScanner {  
    int d1, d2;  
    Scanner sc = new Scanner(System.in);  
    InputScanner() {  
        if (this.getClass() == Circle.class) {  
            System.out.println("Enter d1: ");  
            d1 = sc.nextInt();  
        }  
        else {  
            System.out.println("Enter d1 and d2: ");  
            d1 = sc.nextInt();  
            d2 = sc.nextInt();  
        }  
    }  
}
```

~~abstract class Shape extends InputScanner {  
 abstract void printArea();  
}~~

~~class Triangle extends Shape {  
 void printArea() {  
 System.out.println("Area of triangle is " + (double)  
 (d1 \* d2) / 2);  
 }  
}~~

~~class Rectangle extends Shape {  
 void printArea() {  
 System.out.println("Area of rectangle is : " + (double)  
 (d1 \* d2));  
 }  
}~~

```
class Circle extends Shape {  
    void printArea() {  
        System.out.println("Area of circle " + (double)(3.14 * d1 * d1));  
    }  
  
}  
  
class AreaMain {  
    public static void main (String args[]) {  
        System.out.println ("Rishi - IBM22CS222");  
        Rectangle r = new Rectangle();  
        Triangle tr = new Triangle();  
        Circle c = new Circle();  
        r.printArea();  
        tr.printArea();  
        c.printArea();  
    }  
}
```

### Output -

Enter d1 and d2

5

4

Enter d1 and d2

5

3

Enter d1

6

Area of rectangle is : 20.0

Area of triangle is : 7.5

Area of circle is : 113.039

8  
2/1/24

## Lab - 5

3 math static

7 (Default) class

```

import java.util.*;
class Bank
{
    String name;
    int accno;
    boolean current;
    Scanner sc = new Scanner(System.in);
    Bank()
    {
        if (true.getAcc() == CurrentAcc)
        {
            current = true;
        }
        else
        {
            current = false;
        }
    }
    System.out.println("Enter deposit amount.");
    name = sc.next();
    System.out.println("Enter account no.");
    accno = sc.nextInt();
}

void deposit()
{
    System.out.print("Enter deposit amount");
    balance += sc.nextInt();
}

void withdraw()
{
    System.out.print("Enter deposit amount");
    double withdraw = sc.nextDouble();
    while (withdraw > balance)
    {
        System.out.println("Insufficient Balance");
        withdraw = sc.nextDouble();
    }
}

```

2.0 : is operator  
2.8 : is operator  
2.0 : is show

```

balance -= withdraw;
if (current && balance < min-balance)
    {
        System.out.println ("Below min balance of 100, removing " + remaining money
                            in account);
        balance = 0;
    }
}

void withdraw (double withdraw)
{
    if (withdraw > balance)
        {
            System.out.println ("Withdraw amount greater than balance");
            withdraw = 0;
        }
    if (current && balance < min-balance)
        {
            System.out.println ("Below min balance of 100, removing " + remaining money);
            balance = 0;
        }
}

void showBalance()
{
    System.out.print ("balance = " + balance);
}

class CurrentAcc extends Account
{
    void cheque()
    {
        System.out.print ("Enter cheque amount:");
        double cheque = sc.nextDouble();
        withdraw(cheque);
        System.out.println ("Cheque created");
    }
}

```

class SavingsAcc extends Account {  
void compound (int t, int r) {

{

balance = balance + (Math. pow ((1 + ((double) r / 100)), t));  
System.out.println ("Balance after given rate and time " + balance);

class Bank {

public static void main (String args[]) {

{

SavingsAcc john = new SavingsAcc();

CurrentAcc smith = new CurrentAcc();

Account ref = null;

if (acc == 1)

ref = john; // creates object "john" which is null

ref = & john;

{

else if (acc == 2)

ref = smith;

}

while (choice != 6) {

if (choice == 1) {

ref.deposit(); // creates object "john" which is null

else if (choice == 2)

ref.withdraw();

else if (choice == 3) {

if (acc == 1)

john.compound(1, 5);

else if (choice == 4)

ref.showBalance();

2

System.out.println ("Enter acc no ");

acc = sc.nextInt();

choice = sc.nextInt();

Output -

Enter name : john

Enter acc no : 1

Enter name : Smith

Enter acc\_no : 2

- Choice :  
1) Deposit  
2) Withdraw  
3) Compute interest  
4) Display amount  
5) Create Cheque  
6) Exit

| 5

Enter cheque amount

5000

Cheque created

J S D  
JFS  
09.01.2023

WEEK-6  
LAB-7

23-1-24

Date \_\_\_\_\_

o 1 e 3 f 2 z p e

Student.java

package ie;

public class Student {

    public String name, usn;  
    public int sem;

}

\* eat a lot . that a is lots : prints lots  
eat saw bands . that a bands : prints lots

blowball : prints blowball

swarm : prints lots

Internals.java

package ie;

import java.util.Scanner;

public class Internals extends Student {

public int marks[] = new int [5];

public void InputMarks()

{

Scanner sc = new Scanner (System.in);

for (int i=0; i&lt;5; i++) {

System.out.println ("Enter subject " + (i+1) + " marks");

marks[i] = sc.nextInt();

}

}

bottom bars return sign

bottom bar return draft

bottom full sign

bottom full draft

S, D, P return

+ ?

public void displayMarks()

{

for (int i=0; i&lt;5; i++) {

System.out.println ("Subject " + (i+1) + " marks : " + marks[i]);

}

}

O.P : elegant p all

O.C : elegant p return

O.NE : elegant p all

JOO8.0 : elegant p return



```
finalmarks[i] = new External();  
intmarks[i] = new Internal();  
finalmarks[i] = inputMarks();  
intmarks[i].inputMarks();
```

{

```
for (int i = 0; i < no; i++) {
```

```
System.out.println ("CIE : ");
```

```
intmarks[i].displayMarks();
```

```
System.out.println ("SEE : ");
```

```
finalmarks[i].displayMarks();
```

{

```
; ("Stream" + (i+1) + " higher orbit") ntral - tw . metry
```

```
} subject marks saved in
```

```
{ } tri var = () stream for subject
```

```
{ () stream from book subject
```

```
array var = > array
```

```
{ ++i; i > i; o = i do } ref
```

```
; ( ) HLF from sb = (, ) stream
```

{

### Output -

```
Enter subject 1 marks : 30
```

```
{ () stream publish book subject
```

```
Enter subject 2 marks : 50
```

```
{ ++i; i > i; o = i do } ref
```

```
Enter subject 3 marks : 40 " + (i+1) + " higher orbit" ntral - tw . metry
```

```
Enter subject 4 marks : 20
```

```
Enter subject 5 marks : 10
```

### ~~Subject Marks~~

```
Enter subject 1 marks : 30
```

```
Enter subject 2 marks : 70
```

```
Enter subject 3 marks : 60
```

```
Enter subject 4 marks : 80
```

```
Enter subject 5 marks : 90
```

```
Enter subject 1 marks : 70
```

```
Enter subject 2 marks : 40
```

```
Enter subject 3 marks : 20
```

```
Enter subject 4 marks : 80
```

```
Enter subject 5 marks : 10
```

CIE:

- Subject 1 marks : 30  
 Subject 2 marks : 50  
 Subject 3 marks : 40  
 Subject 4 marks : 20  
 Subject 5 marks : 10

SEE-

- Subject 1 marks : 30  
 Subject 2 marks : 70  
 Subject 3 marks : 60  
 Subject 4 marks : 80  
 Subject 5 marks : 90

CIE-

- Subject 1 marks : 70  
 Subject 2 marks : 40  
 Subject 3 marks : 20  
 Subject 4 marks : 20  
 Subject 5 marks : 10

SEE-

- Subject 1 marks : 80  
 Subject 2 marks : 60  
 Subject 3 marks : 30  
~~Subject 4 marks : 20~~  
~~Subject 5 marks : 30~~

~~23 01 2011~~

(~~"s'effacer et faire apparaître"~~) afficher une variété de styles  
 (~~"s'effacer et faire apparaître"~~) afficher des motifs

## Week - 7

import java.util.Scanner;  
class WrongAge extends Exception {  
 WrongAge (String error) {  
 System.out.println (error);  
 }  
}

class Father {  
int age;

Father (int age) throws Wrong Age {  
if (age < 0)

~~-throw new WrongAge ("Father's age cannot be negative");~~  
~~this . age = age;~~

}  
} class Son extends Father {  
int age;  
Son (int age, int s-age) throws WrongAge {  
super(age);

~~if ( $S - \text{age} \geq \text{age}$ )  
throw new WrongAge ("Son's age cannot be greater than Father's age");  
 $\text{this.age} = S - \text{age};$~~

```
    }  
    }  
class Main {
```

```
public static void main ( String args [] ) {  
    Scanner sc = new Scanner ( System . in );  
    try {  
        System . out . println ( " Enter the Father's age " );
```

```
int f-age = sc.nextInt();  
System.out.println("Enter the son's age");  
int s-age = sc.nextInt();  
Son a = new Son(f-age, s-age);  
System.out.println("Father's age : " + f-age);  
System.out.println("Son's age " + s-age);  
}
```

```
catch (WrongAge e) {
```

```
System.out.println("Wrong Age entered"); }
```

```
catch (Exception ee) {
```

```
System.out.println("Unexpected error : " + ee); }
```

```
}
```

```
}
```

## OUTPUT-1

Enter the Father's Age

45

Enter the Son's Age

15

Father's age - 45

Son's Age - 15

## OUTPUT-2

Enter Father's age

-21

Father's age cannot be negative

~~SCV  
SNS  
30.01.24~~

## lab - 10

class A {

    synchronized void foo(B b) { if ("that - A start") allowed. two. not synchronized

        String name = Thread.currentThread().getName();

        System.out.println(name + " entered A.foo");

    try {

        Thread.sleep(1000);

    } catch (Exception e) {

        System.out.println(" A interrupted");

}

; ("barred wait") small code - () barrier(Thread). barrier  
System.out.println(name + " trying " + " barrier wait = & barred");

    b.last();

}

void last() {

; ("barred wait is dead") allowed. two. not synchronized

```
System.out.println("Inside (A.last)"); // from bnew state, since  
; () B.wrt = d B  
; () C.wrt = 0 )  
; () true . d  
; () false . 0  
class B {
```

Synchronised wait bar (A a) {

```
String name = Thread.currentThread().getName();
```

System.out.println ( name + " entered B. hrs ");

-by {

Thread. Sleep (1000);

} catch (Exception e) {

System.out.println ("B Interrupted");

۳

```
System.out.println( name + " trying to call A.last() " );
```

a. last();

3

```
void last() {
```

System.out.println ("Inside A . last"); (d 8) sof bzw. beispielsweise  
 ; ()smaltp. ()kauft stroh - kauft = smar . print()  
 ; (" sof . A . berline " + smar ) ntwurf . fws . mctyp

class Deadlock implements Runnable {

A a = new A();

B b = new B();

Doddle 123

~~Thread. currentThread().setName("Main Thread");~~

~~Thread t = new Thread (this, "Racing Thread"); t.start();~~

t. start();

a. `foo(b);`

System.out.println ("Back in main thread");

۳

```
public static void main (String args[]) { // main. this. method  
    new Deadlock();  
}  
} // class Deadlock
```

## OUTPUT -

Main Thread entered A.foo

Racing Thread entered B.bar

Main Thread trying to call B.last()  
Inside A.last

Back in main Thread

Racing Thread trying to call A.last() // (S) attempt. this. method

Inside A.last

Back in other thread

ANS 02. 24  
06.

## Proton - Java

class Q {

int n;

boolean valueSet = false;

synchronized int get() {

while (!valueSet) {

try {

(\*) deals with "deadlock", if it's been taken

System.out.println("Consumed Waiting");

wait();

} catch (InterruptedException e) {

System.out.println(e); }

System.out.println(" " + n); } from line 10th slide  
value set = false; Got: ; 1st part was

System.out.println(" Tell producer");  
notify();

}

return n;

}

synchronized void put(int n) {

while (value set) {

try {

System.out.println(" Producer Waiting.");  
wait();

} catch (InterruptedException e) {

System.out.println(e);

}

this.n = n;

value set = true;

System.out.println("Put: " + n);

System.out.println(" Tell consumer");

notify();

W.B.  
V.S., J.G.

NO

class Producer implements Runnable {

Q q;

Producer(Q q) {

this.q = q;

new Thread(this, "Producer").start();

}

int i = 0;

while (i < 3) {

initial = did writer anything

? (I step into debugger here)

? (de-mutate!) click

? wt

? (initial demand) retrieve start-method?

? (I fix it)

{ (setfirstbefore) start {  
{ i(0).retrieve.start.method

```

    q.put(i++);
}
}

class Consumer implements Runnable {
    Q q;
    consumer(Q q) {
        this.q = q;
        new Thread(this, "Consumer").start();
    }
    public void run() {
        int i = 0;
        while (i < 3) {
            int n = q.get();
            System.out.println("Consumed : " + i);
            i++;
        }
    }
}

```

### OUTPUT-1

Put : 1

Get : 1

Put : 2

Get : 2

Put : 3

Get : 3

13/2/2018

Lab - 8

Week - 2

import java.util.\*;

class B extends Thread {

public void run() {

try {

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

System.out.println("BMS");

Thread.sleep(1000);

}

} catch (InterruptedException e) {

System.out.println(e);

}

3

class C extends Thread {

public void run() {

try {

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

System.out.println("CSE");

Thread.sleep(2000);

}

} catch (InterruptedException e) {

System.out.println(e);

3  
3

class ThreadMain {

```
public static void main ( String args [] ) { blank " ) without . this . method  
B b = new B ();  
C c = new C ();  
b. start ();  
c. start ();  
}  
} ( > A ) start from below  
; () small step. () build throw . hardt = main with  
; (" say . I . build " + main ) without . this . method  
} yet
```

## OUTPUT-

BMS

CSE

CSE

CSE

BMS

BMS

```
; (001) feels . hardt  
} ( > nextpart ) dids {  
; (" buildfirst 8 " ) without . this . method  
} (111) . 2
```

## LAB-9

```
import javax.swing.*;  
import java.awt.*;  
import java.awt.event.*;  
  
class SwingDemo {  
    SwingDemo() {  
        JFrame jfrm = new JFrame ("Divide App");  
        jfrm.setSize (275, 150);  
        jfrm.setLayout (new FlowLayout());  
        jfrm.setDefaultCloseOperation (JFrame.EXIT_ON_CLOSE);  
        JLabel jlab = new JLabel ("Enter the divisor and dividend");  
        JTextField ejtf = 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 (ejtf);  
        jfrm.add (bjtf);  
        jfrm.add (button);  
        jfrm.add (alab);  
        jfrm.add (blab);  
        jfrm.add (anslab);  
    }  
}
```

```

Action Listener 1 = new ActionListener() {
    public void actionPerformed(ActionEvent evt) {
        System.out.println("Action event from a text field");
    }
};

if . add ActionListener(0); // (0) and plus was
if . add ActionListener(1); // (1)

button . add ActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent evt) {
        try {
            int a = Integer.parseInt(if . getText()); - TURNO
            int b = Integer.parseInt(if . getText()); - button was
            int ans = a/b;
            alab . setText("In A = " + a);
            blab . setText("In B = " + b); // A = A
            anlab . setText("In Ans = " + ans);
        }
    }
});

catch (NumberFormatException e) { - translate draft
    alab . setText(".");
    blab . setText(".");
    anlab . setText("0"); // without reg below draft - tree - draft
    err . setText("Enter only integers!"); - below
}

catch (ArithmeticException e) { - number
    alab . setText("0"); // for zero - not draft
    blab . setText("0"); // needed negative - not draft
    anlab . setText("0"); // negative text because A - negative
    err . setText("B should be Non zero!"); // not - neither
}
}

```

```

    item.setVisible(true); } () verhindert war + ! verhindert
}
] (true true nicht) bewirkt nichts bzw. nichts
public static void main (String args []) {
    SwingUtilities.invokeLater (new Runnable) {
        public void run () {
            new Swing Demo (); ; () verhindert nichts bzw. nichts
        }
    });
}
} (true true nicht) bewirkt nichts bzw. nichts
}

```

### OUTPUT -

i((1) true tag - iffe bz ) true swang . vegetat = o tri

Enter (divider tag - diff) true stat . vegetat = d tri

120

30

i((e + " = A n/ ") true tag . date

Calculate

A = 120

i((f + 30" = Ans = 6, ") true tag . date

i((ans + " = ans n/ ") true tag . date

### Explanation -

#### Import statement -

java . swing - imports from Swing toolkit (i.e.) graphics user interface

java . awt - imports from abstract window toolkit providing data

java . event - imports classes for handling button click events.

#### Classes -

Swing Demo - Main class for defining the app's logic.

JFrame - A top level window container

JLabel - A non editable text label to disp info

JTextField - User input

JButton - Button that triggers actions.

FlowLayout - A manager that arranges components in horizontal flow.

Action Listener - An interface for handling action events.

## Functions -

- new JFrame ("Divider App") ; new JFrame with specified title .
  - set Size (275 , 150) - Sets the initial size of the frame .
  - setLayout - Sets layout manager to FlowLayout.
  - setDefaultCloseOperation - Terminates when frame is closed.
- new JLabel - Creates a label with the given text
- JTextField - Creates field with 8 columns for input.
- JButton - Creates button with the given text.

Action Listener - Attached to component to handle action events .

ActionEvent - The method called when action event occurs.

Swing Utilities - Schedules a runnable task to be executed on EDT .

*Java GUI Programming*

## LAB-1

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 discriminant  $b^2 - 4ac$  is negative, display a message stating that there are no real solutions.

```
import java.util.Scanner;

class Quadratic {
    int a, b, c;
    double r1, r2, d;

    void getCoefficients() {
        Scanner scanner = new Scanner(System.in);
        System.out.println("Enter the coefficients of a, b, c:");
        a = scanner.nextInt();
        b = scanner.nextInt();
        c = scanner.nextInt();
        scanner.close();
    }

    void computeRoots() {
        while (a == 0) {
            System.out.println("Not a quadratic equation");
            System.out.println("Enter a non-zero value for a:");
            Scanner scanner = new Scanner(System.in);
            a = scanner.nextInt();
            scanner.close();
        }

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

        if (d == 0) {
            r1 = -b / (2.0 * a);
            System.out.println("Roots are real and equal");
            System.out.println("Root1 = Root2 = " + r1);
        } else if (d > 0) {
            r1 = (-b + Math.sqrt(d)) / (2.0 * a);
            r2 = (-b - Math.sqrt(d)) / (2.0 * a);
            System.out.println("Roots are real and distinct");
            System.out.println("Root1 = " + r1 + " Root2 = " + r2);
        } else if (d < 0) {
            System.out.println("No real roots");
        }
    }
}
```

```
        System.out.println("Roots are imaginary");
        r1 = -b / (2.0 * a);
        r2 = Math.sqrt(-d) / (2.0 * a);
        System.out.println("Root1 = " + r1 + " + i" + r2);
        System.out.println("Root2 = " + r1 + " - i" + r2);
    }
}

class QuadraticMain {
    public static void main(String args[]) {
        Quadratic quadraticEquation = new Quadratic();
        quadraticEquation.getCoefficients();
        quadraticEquation.computeRoots();
    }
}
```

Output:

```
; if (?) { javac QuadraticMain.java } ; if (?) { java QuadraticMain }
Enter the coefficients of a, b, c:
1 2 3
Roots are imaginary
Root1 = -1.0 + i1.4142135623730951
Root2 = -1.0 - i1.4142135623730951
```

## LAB-2

**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,name;
    int n;
    String subject[]=new String[10];
    int credits[]=new int[10];
    int marks[]=new int[10];

    void read(){
        Scanner input= new Scanner(System.in);

        System.out.print("Enter your name:");
        name=input.nextLine();

        System.out.print("\n");
        System.out.print("Enter your USN:");
        usn=input.nextLine();
        System.out.print("\n");

        System.out.print("Enter number of subjects:");
        n=input.nextInt();

        input.nextLine();

        System.out.print("\n");
        for(int i=0;i<n;i++){
            System.out.println(" -----");
            System.out.print("Enter subject:");
            subject[i]=input.nextLine();

            System.out.print("\n");
        }
    }

    void display(){
        System.out.print("-----");
        System.out.print("USN: "+usn);
        System.out.print("Name: "+name);
        System.out.print("Number of Subjects: "+n);
        System.out.print("Subject Details: ");
        for(int i=0;i<n;i++){
            System.out.print(subject[i]);
            System.out.print(" -----");
        }
        System.out.print("Total Credits: "+credits);
        System.out.print("SGPA: "+(float)credits/n);
    }
}
```

```
System.out.print("Enter credits:");credits[i]=input.nextInt();

System.out.print("\n");

System.out.print("Enter Marks:");marks[i]=input.nextInt();

System.out.print("\n"); System.out.println(" ----- ");
input.nextLine();

}

}

void display(){

System.out.println("-----");

System.out.println("Name:"+name); System.out.println("USN:"+usn+"\n");

System.out.println("Subject\tCredits\tMarks");
System.out.println("-----");
for(int i=0;i<n;i++){

System.out.println(subject[i]+\t+credits[i]+\t+marks[i]);
}

System.out.println("-----");

int grade(int a){

while(a/10!=0){
```

```
a=a/10;
}
if((a+1)>5){
    return (a+1);
}
else if((a+1)==5){
    return a;
}
else{return 0;}

}

void calculate(){

int total_credits=0;int
marks_credits=0;

for(int i=0;i<n;i++){
    total_credits+=credits[i]; marks_credits+=grade(marks[i])*credits[i];
}

System.out.println("SGPA is
"+(double)marks_credits/total_credits);
}

}

class marks{
    public static void main(String[] args){

        student calc= new student();
        calc.read(); calc.display();
        calc.calculate();

    }
}
```

Subject	Credits	Marks
---------	---------	-------

Math	4	98
Java	4	89
DBMS	3	99
COA	3	88
Kannada	1	98

SGPA is 9.533333333333333
---------------------------

### LAB-3

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 books{
    String author,book;
    int price,numPages;

    books(String book,String author, int price, int numPages){
        this.book=book;
        this.author=author;
        this.price=price;
        this.numPages=numPages;
    }

    public String toString(){
        return "Book
Name:"+this.book+"\n"+ "Author:"+this.author+"\n"+ "Price:"+this.price+"\n"+
Number of pages:"+this.numPages+"\n";
    }

}

class lib{
```

```
public static void main(String[] args){String
    author,book;
    int price,numPages;int

    num_book;

    Scanner input=new Scanner(System.in);

    System.out.print("Enter number of books:");
    num_book=input.nextInt(); System.out.print("\n");

    books b[]={new books[num_book];for(int

        i=0;i<num_book;i++){
        System.out.println("----- \nBook
        "+(i+1)+"\n-----");
        input.nextLine();
        System.out.print("Enter the name of book:");book=input.nextLine();

        System.out.print("Enter Authors name:");author=input.nextLine();

        System.out.print("Enter the price:");price=input.nextInt();

        System.out.print("Enter the number of pages:");numPages=input.nextInt();

        b[i]=new books(book,author,price,numPages);

    }

    for(int i=0;i<num_book;i++){
        System.out.println("-----\nBook "+(i+1)+"details\n-----");
    }
}
```

```
        String a=b[i].toString();
        System.out.println(a);

    }

}

}
```

### Output:

```
Book 1
-----
Enter the name of book:Harry Potter
Enter Authors name:JK Rowling
Enter the price:799
Enter the number of pages:889
-----
Book 2
-----
Enter the name of book:Percy Jackson
Enter Authors name:Rick Riordan
Enter the price:689
Enter the number of pages:459
-----
Book 3
-----
Enter the name of book:Goosebumps
Enter Authors name:RL Stine
Enter the price:450
Enter the number of pages:236
-----
Book 1 details
-----
Book Name:Harry Potter
Author:JK Rowling
Price:799
Number of pages:889
-----
Book 2 details
-----
Book Name:Percy Jackson
Author:Rick Riordan
Price:689
Number of pages:459
-----
Book 3 details
-----
Book Name:Goosebumps
Author:RL Stine
Price:450
Number of pages:236
```



#### **LAB-4**

**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 classShape. Each one of the classes contain only the method printArea( ) that prints the area of the given shape.**

```
import java.util.Scanner;

class inputScanner{
    void rec(rectangle ab){
        Scanner input=new Scanner(System.in);
        System.out.println("Enter the dimensions of the
rectangle(Length and Breadth):");
        ab.a=input.nextInt();
        ab.b=input.nextInt();
    }

    void tri(triangle ab){

        Scanner input=new Scanner(System.in);
        System.out.println("Enter the dimensions of the triangle(base
and height):");
        ab.a=input.nextInt();
        ab.b=input.nextInt();
    }

    void cir(circle ab){
        Scanner input=new Scanner(System.in);
        System.out.println("Enter the dimension of the
circle(radius):");
        ab.a=input.nextInt();
    }

}

abstract class shape extends inputScanner{
    int a,b;
    abstract void printArea();
}
```

```
class rectangle extends shape{
    rectangle(){}
        rec(this);
    }
    void printArea(){
        System.out.println("Area of Rectangle = "+(double)(a*b));
    }
}

class triangle extends shape{
    triangle(){tri(this);}
    void printArea(){
        System.out.println("Area of Triangle = "+(0.5*a*b));
    }
}

class circle extends shape{
    circle(){cir(this);}
    void printArea(){
        System.out.println("Area of Circle = "+(3.14*a*a));
    }
}

class calc{
    public static void main(String[] args){
        rectangle r=new rectangle();
        triangle t=new triangle();
        circle c=new circle();

        r.printArea();
        t.printArea();
        c.printArea();
    }
}
```

**Output:**

```
Enter the dimensions of the rectangle(Length and Breadth):  
4 5  
Enter the dimensions of the triangle(base and height):  
9 8  
Enter the dimension of the circle(radius):  
4  
Area of Rectangle = 20.0  
Area of Triangle = 36.0  
Area of Circle = 50.24
```

## LAB-5

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 name;  
    int accountNumber;  
    double balance = 0;  
    Scanner scanner = new Scanner(System.in);  
  
    Account(String accountType) {  
        System.out.println("Creating a new " + accountType + " account");  
        System.out.print("Enter name: ");
```

```
this.name = scanner.next(); System.out.print("Enter  
account number: ");this.accountNumber =  
scanner.nextInt();  
}  
  
void deposit() {  
    System.out.print("Enter deposit amount: ");balance +=  
    scanner.nextDouble();  
    System.out.println("Deposit successful. Current balance: " +balance);  
}  
  
void withdraw() {  
    System.out.print("Enter withdrawal amount: ");double  
    withdrawal = scanner.nextDouble();  
    if (withdrawal <= balance) {balance -=  
        withdrawal;  
        System.out.println("Withdrawal successful. Current balance: " +  
balance);  
    } else {  
        System.out.println("Insufficient funds for withdrawal");  
    }  
}  
  
void displayBalance() {  
    System.out.println("Current balance: " + balance);  
}  
  
void performMenuActions() {int  
    choice;  
    do {  
        System.out.println("\n----- MENU -----");  
        System.out.println("1. Deposit");  
        System.out.println("2. Withdraw");  
        System.out.println("3. Display Balance");  
        System.out.println("4. Exit");  
        System.out.print("Enter your choice: ");choice =  
        scanner.nextInt();  
  
        switch (choice) {case  
            1:  
                deposit();  
                break;  
        }  
    }  
}
```

```

        case 2:
            withdraw();
            break;
        case 3:
            displayBalance();
            break;
        case 4:
            System.out.println("Exiting the menu. Thank you!");break;
        default:
            System.out.println("Invalid choice. Please try
again.");
        }
    } while (choice != 4);
}
}

class SavingsAccount extends Account {int
interestRate = 5;

SavingsAccount() {
    super("Savings");
}

void compoundInterest(int time) {
    balance *= Math.pow((1 + (interestRate / 100.0)), time); System.out.println("Compound
interest applied. Current balance: " +
balance);
}
}

class CurrentAccount extends Account {double
overdraftLimit = -100;

CurrentAccount() {
    super("Current");
}

void issueCheque() {
    System.out.print("Enter cheque amount: ");double
    chequeAmount = scanner.nextDouble();
    if (chequeAmount <= balance && (balance - chequeAmount) >=overdraftLimit) {

```

```
        balance -= chequeAmount;
        System.out.println("Cheque issued successfully. Currentbalance: " +
balance);
    } else {
        System.out.println("Insufficient funds to issue the cheque");
    }
}

public class bank {
    public static void main(String[] args) { Scanner scanner =
        new Scanner(System.in);

    System.out.println("Welcome to the Banking App");

    SavingsAccount savingsAccount = new SavingsAccount();
    CurrentAccount currentAccount = new CurrentAccount();

    Account selectedAccount = null;

    System.out.println("\nSelect an account type:");
    System.out.println("1. Savings Account"); System.out.println("2.
Current Account");
    int accountTypeChoice = scanner.nextInt();

    if (accountTypeChoice == 1) { selectedAccount =
        savingsAccount;
    } else if (accountTypeChoice == 2) { selectedAccount =
        currentAccount;
    } else {
        System.out.println("Invalid choice. Exiting.");System.exit(0);
    }

    selectedAccount.performMenuActions();

    scanner.close();
}
}
```

**Output:**

```
Welcome to the Banking App
Creating a new Savings account
Enter name: Smith
Enter account number: 1
Creating a new Current account
Enter name: John
Enter account number: 2

Select an account type:
1. Savings Account
2. Current Account
1

----- MENU -----
1. Deposit
2. Withdraw
3. Display Balance
4. Exit
Enter your choice: 1
Enter deposit amount: 8000
Deposit successful. Current balance: 8000.0

----- MENU -----
1. Deposit
2. Withdraw
3. Display Balance
4. Exit
Enter your choice: 2
Enter withdrawal amount: 700
Withdrawal successful. Current balance: 7300.0

----- MENU -----
1. Deposit
2. Withdraw
3. Display Balance
4. Exit
Enter your choice: 3
Current balance: 7300.0
```

## LAB-6

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/Student.java

```
package CIE;

public class Student {
    public String usn;
    public String name;
    public int sem;

    public Student(String usn, String name, int sem) {
        this.usn = usn;
        this.name = name;
        this.sem = sem;
    }
}

// CIE/Internals.java
package CIE;

public class Internals extends Student {
    public int[] internalMarks;

    public Internals(String usn, String name, int sem, int[] internalMarks)
    {
        super(usn, name, sem);
        this.internalMarks = internalMarks;
    }
}
```

### **SEE/External.java**

```
package SEE;

import CIE.Student;

public class External extends Student {
    public int[] seeMarks;

    public External(String usn, String name, int sem, int[] seeMarks) {
        super(usn, name, sem);
        this.seeMarks = seeMarks;
    }
}
```

### **FinalMarks.java**

```
import CIE.Internals;
import SEE.External;
import java.util.Scanner;

public class FinalMarks {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter the number of students: ");
        int n = scanner.nextInt();

        CIE.Internals[] cieStudents = new CIE.Internals[n];
        SEE.External[] seeStudents = new SEE.External[n];

        for (int i = 0; i < n; i++) {
            System.out.println("Enter details for CIE of student " + (i + 1));
            cieStudents[i] = new CIE.Internals(usn, name, sem, cieMarks);
        }

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

```
        System.out.println("Enter details for SEE of student " + (i +
1));

        seeStudents[i] = new SEE.External(usn, name, sem, seeMarks);
    }

    System.out.println("\nFinal Marks of Students:");
    for (int i = 0; i < n; i++) {
        System.out.println("\nDetails of Student " + (i + 1));

    }
}
```

## LAB-7

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 25 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;
import java.lang.Exception;

class WrongAge extends Exception{
    WrongAge(String s){
        super(s);
    }
}

class InputScanner {
    Scanner sc = new Scanner(System.in);
    int Age;
    InputScanner(){
        if (this.getClass() == Father.class){
            System.out.println("Enter father age: ");
            Age = sc.nextInt();
        }
    }
}

class Father extends InputScanner{
    int FatherAge;
    Father() throws WrongAge{
        FatherAge = Age;
        if (FatherAge < 0){
            throw new WrongAge("Age cannot be < 0 for a person");
        }
    }
    void display(){
        System.out.println("Father Age = " + FatherAge);
    }
}

class Son extends Father{
    int SonAge;
    Son() throws WrongAge{
        super();
        System.out.println("Enter son age: ");
    }
}
```

```

SonAge = sc.nextInt();
if (FatherAge < SonAge){
    throw new WrongAge("Age cannot be greater for son");
} else if (SonAge < 0){
    throw new WrongAge("Age cannot be < 0 for a person");
}
}

void display(){
    System.out.println("Son Age = " + SonAge);
}

}

class agemain{
    public static void main(String[] args) {
        try{
            Father father = new Father();
            Son son = new Son();
            father.display();
            son.display();
        }
        catch (WrongAge e){
            System.out.println(e.getMessage());
        }
    }
}

```

### Output:

```

Enter father's age: 54
Enter son's age: 21
Father's age: 54
Son's age: 21
PS C:\Users\rabhi\OneDrive\Desktop\Java\lab> cd "c:\Users\rabhi\OneDrive\Desktop\Java\lab"
\>; if ($?) { javac Main.java } ; if (?) { java Main }
Enter father's age: 21
Enter son's age: 54
Exception caught: WrongAge: Son's age cannot be greater than or equal to father's age
Exception caught: Son's age cannot be greater than or equal to father's age

```

## LAB-8

**Write a program which creates two threads, one thread displaying “BMS Collegeof Engineering” once every ten seconds and another displaying “CSE” once every two seconds.**

```
import java.io.*;

class B extends Thread{
    public void run(){
        try{
            for(int i = 0; i < 3; i++){
                System.out.println("BMS");
                Thread.sleep(10000);
            }
        } catch (InterruptedException e){
            System.out.println(e);
        }
    }
}

class C extends Thread{
    public void run(){
        try{
            for(int i = 0; i < 3; i++){
                System.out.println("CSE");
                Thread.sleep(2000);
            }
        } catch (InterruptedException e){
            System.out.println(e);
        }
    }
}

class ThreadMain{
    public static void main(String args[]){
        B b = new B();
        C c = new C();
```

```
        b.start();
        c.start();
    }
}
```

## Output

```
BMS
CSE
CSE
CSE
BMS
BMS
```

## LAB-9

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("Calculate");

// 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("Action event from a text
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;

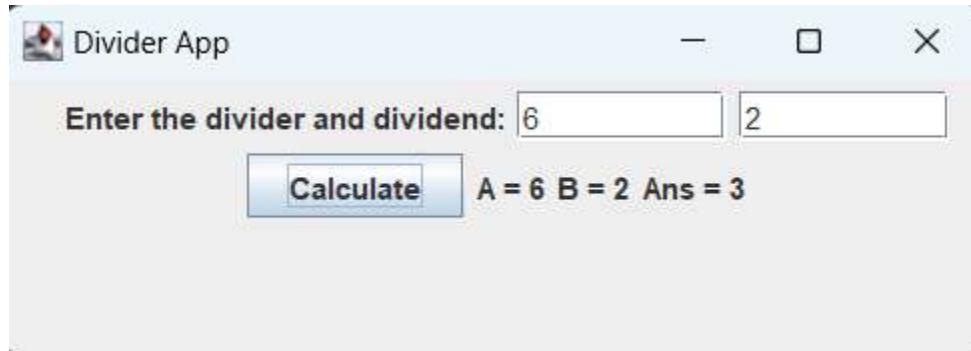
        alab.setText("\nA = " + a);
        blab.setText("\nB = " + b);
        anslab.setText("\nAns = " + ans);
    } catch (NumberFormatException e) {
        alab.setText("");
        blab.setText("");
        anslab.setText("");
        err.setText("Enter Only Integers!");
    } catch (ArithmaticException e) {
        alab.setText("");
        blab.setText("");
        anslab.setText("");
        err.setText("B should be NON zero!");
    }
}
```

```
        }
    });

    // 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();
        }
    });
}
}
```

## Output



## LAB-10

Demonstrate Inter process Communication and deadlock.

deadlock.java

```
class A {  
    synchronized void foo(B b) {  
        String name = Thread.currentThread().getName();  
        System.out.println(name + " entered A.foo"); try {  
            Thread.sleep(1000);  
        } catch (Exception e) { System.out.println("A  
                Interrupted");  
        }  
        System.out.println(name + " trying to call B.last()");b.last();  
    }  
  
    void last() {  
        System.out.println("Inside A.last");  
    }  
}  
  
class B {  
  
    synchronized void bar(A a) {  
        String name = Thread.currentThread().getName();  
        System.out.println(name + " entered B.bar"); try {  
            Thread.sleep(1000);  
        } catch (Exception e) { System.out.println("B  
                Interrupted");  
        }  
  
        System.out.println(name + " trying to call A.last()");a.last();  
    }  
  
    void last() {  
        System.out.println("Inside A.last");  
    }  
}
```

```
    }

}

class Deadlock implements Runnable {
    A a = new A();
    B b = new B();

    Deadlock() {
        Thread.currentThread().setName("MainThread");
        Thread t = new Thread(this, "RacingThread");
        t.start();
        a.foo(b); // get lock on a in this thread.
        System.out.println("Back in main thread");
    }

    public void run() {
        b.bar(a); // get lock on b in other thread.
        System.out.println("Back in other thread");
    }

    public static void main(String args[]) {
        new Deadlock();
    }
}
```

## Output

```
MainThread entered A.foo
RacingThread entered B.bar
MainThread trying to call B.last()
Inside A.last
RacingThread trying to call A.last()
Back in main thread
Inside A.last
Back in other thread
```

```
procon.java
```

```
import java.lang.*;class  
  
Q {  
  
    int n;  
  
    boolean valueSet = false;  
  
    synchronized int get() {  
  
        while (!valueSet) {try {  
            System.out.println("\nConsumer waiting\n");wait();  
  
        } catch (InterruptedException e) { System.out.println("InterruptedException  
caught");}  
        }  
        System.out.println("Got: " + n);valueSet =  
        false;  
        System.out.println("\nIntimate Producer\n");notify();  
    }  
  
    return n;  
}  
  
synchronized void put(int n) {  
  
    while (valueSet) {try {  
        System.out.println("\nProducer waiting\n");wait();  
  
    } catch (InterruptedException e) { System.out.println("InterruptedException  
caught");}  
    }  
    this.n = n; valueSet =  
    true;  
    System.out.println("Put: " + n); System.out.println("\nIntimate  
Consumer\n");notify();  
}
```

```
        }
    }
}

class Producer implements Runnable {Q q;

    Producer(Q q) {
        this.q = q;
        new Thread(this, "Producer").start();
    }

    public void run() { int i =
        0; while (i < 15) {
            q.put(i++);
        }
    }
}

class Consumer implements Runnable {Q q;

    Consumer(Q q) {
        this.q = q;
        new Thread(this, "Consumer").start();
    }

    public void run() { int i =
        0; while (i < 15) {
            int r = q.get(); System.out.println("consumed:" +
            r);i++;
        }
    }
}

class ProCon {
    public static void main(String args[]) {Q q = new Q();
        new Producer(q);
        new Consumer(q);
    }
}
```

```
    }  
}
```

## Output

```
Intimate Consumer  
  
Producer waiting  
Got: 0  
Intimate Producer  
Put: 1  
Intimate Consumer  
  
Producer waiting  
consumed:0  
Got: 1  
Intimate Producer  
consumed:1  
Put: 2  
Intimate Consumer  
  
producer waiting  
Got: 2  
Intimate Producer  
consumed:2  
Put: 3  
Intimate Consumer  
  
producer waiting  
Got: 3  
Intimate Producer  
consumed:3  
Put: 4  
Intimate Consumer  
  
producer waiting  
Got: 4  
Intimate Producer  
consumed:4  
Put: 5  
Intimate Consumer
```

```
Producer waiting
```

```
Got: 10
```

```
Intimate Producer
```

```
consumed:10
```

```
Put: 11
```

```
Intimate Consumer
```

```
Producer waiting
```

```
Got: 11
```

```
Intimate Producer
```

```
consumed:11
```

```
Put: 12
```

```
Intimate Consumer
```

```
Producer waiting
```

```
Got: 12
```

```
Intimate Producer
```

```
consumed:12
```

```
Put: 13
```

```
Intimate Consumer
```

```
Producer waiting
```

```
Got: 13
```

```
Intimate Producer
```

```
consumed:13
```

```
Put: 14
```

```
Intimate Consumer
```

```
Got: 14
```

```
Intimate Producer
```

```
consumed:14
```