

**VISVESVARAYA TECHNOLOGICAL UNIVERSITY**  
“JnanaSangama”, Belgaum -590014, Karnataka.



**LAB REPORT**  
**on**  
**Object Oriented Java Programming**  
**(23CS3PCOOJ)**

*Submitted by*

**ARCHITA V (1BM23CS050)**

*in partial fulfillment for the award of the degree of*  
**BACHELOR OF ENGINEERING**  
*in*  
**COMPUTER SCIENCE AND ENGINEERING**



**B.M.S. COLLEGE OF ENGINEERING**  
(Autonomous Institution under VTU)  
**BENGALURU-560019**  
**Sep-2024 to Jan-2025**

**B.M.S. College of Engineering,  
Bull Temple Road, Bangalore 560019**  
(Affiliated To Visvesvaraya Technological University, Belgaum)  
**Department of Computer Science and Engineering**



**CERTIFICATE**

This is to certify that the Lab work entitled “Object Oriented Java Programming (23CS3PCOOJ)” carried out by ARCHITA V (**1BM23CS050**), who is a bonafide student of **B.M.S. College of Engineering**. It is in partial fulfillment for the award of **Bachelor of Engineering in Computer Science and Engineering** of the Visvesvaraya Technological University, Belgaum. The Lab report has been approved as it satisfies the academic requirements in respect of an Object Oriented Java Programming (23CS3PCOOJ) work prescribed for the said degree.

Dr Prasad G R Professor Department of CSE, BMSCE	Dr. Jyothi S Nayak Professor & HOD Department of CSE, BMSCE
--	---

## Index

<b>Sl. No.</b>	<b>Date</b>	<b>Experiment Title</b>	<b>Page No.</b>
1	09/10/2024	Quadratic Equation	5-6
2	16/10/2024	Calculate sgpa of a student	7-9
3	23/10/2024	Creation of n book object	10-11
4	23/10/2024	Abstract Class-Shape	12-13
5	30/10/2024	Class Bank	14-16
6	13/11/2024	CIE Package	17-20
7	20/11/2024	Exception handling	21-23
8	27/11/2024	Threads	24-26
9	27/11/2024	Interface	27-29
10	27/11/2024	Inter process communication and Deadlock	30-35

**Github Link:**

<https://github.com/1BM23CS050/JAVA->

## 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.

### ALGORITHM:

09.10.24  
Week Lab - 1

1 Develop a java program that prints all the real solutions to quadratic equation  $ax^2 + bx + c = 0$ . Read in a,b,c and use 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 {  
    public static void main(String[] strings) {  
        Scanner input = new Scanner(System.in);  
        System.out.println("Enter a : ");  
        double a = input.nextDouble();  
        System.out.println("Enter b : ");  
        double b = input.nextDouble();  
        System.out.println("Enter c : ");  
        double c = input.nextDouble();  
        double d = b * b - 4.0 * a * c;  
        if (d > 0.0) {  
            double r1 = (-b + Math.sqrt(d)) / (2.0 * a);  
            double r2 = (-b - Math.sqrt(d)) / (2.0 * a);  
            System.out.println("The roots are " + r1 + " and " + r2);  
        }  
        else if {  
            double r1 = -b / (2.0 * a);  
            System.out.println("Root is " + r1);  
        }  
        else {  
            System.out.println("Roots are not real");  
        }  
    }  
}
```

Output- Enter a : 1  
Enter b : 2  
Enter c : 1  
The root is 1.0

CODE:

```
import java.util.Scanner;
class quadratic1
{
public static void main(String[] Strings)
{
Scanner input = new Scanner(System.in);
System.out.print("Enter the value of a: ");
double a = input.nextDouble();
System.out.print("Enter the value of b: ");
double b = input.nextDouble();
System.out.print("Enter the value of c: ");
double c = input.nextDouble();
double d= b * b - 4.0 * a * c;
if (d> 0.0)
{
double r1 = (-b + Math.pow(d, 0.5)) / (2.0 * a);
double r2 = (-b - Math.pow(d, 0.5)) / (2.0 * a);
System.out.println("The roots are " + r1 + " and " + r2);
}
else if (d == 0.0)
{
double r1 = -b / (2.0 * a);
System.out.println("The root is " + r1);
}
else
{
System.out.println("Roots are not real.");
System.out.println("Archita V 1BM23CS050");
}
}
}
```

```
C:\Users\bmsce\Documents\1BM23CS050>java quadratic1
Enter the value of a: 1
Enter the value of b: -2
Enter the value of c: 1
The root is 1.0
```

## 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.

### ALGORITHM:

```

o 16 10:24
Lab-02
Develop a java program to calculate sgpa.

1 import java.util.*;
class Stud_det {
    int m[] = new int[8];
    int c[] = new int[8];
    int p[] = new int[8];
    int sum;
    String name;
    double sgpa;
    Scanner s = new Scanner(System.in);
    void getdetails () {
        System.out.println("Enter name : ");
        name = s.next();
        System.out.println("Enter usn : ");
        usn = s.next();
        for (int i=0; i<8; i++) {
            System.out.print("Enter marks of subject " + (i+1) + ": ");
            m[i] = s.nextInt();
            System.out.print("Enter credits for subject " + (i+1));
            c[i] = s.nextInt();
        }
        void gradepoint () {
            for (int i=0; i<8; i++)
            if (m[i]>=90 && m[i]<=100)
                p[i]=10;
            else if (m[i]>=80 && m[i]<=90)
                p[i]=9;
            else if (m[i]>=70 && m[i]<=80)
                p[i]=8;
            else if (m[i]>=60 && m[i]<=70)
                p[i]=7;
            else if (m[i]>=50 && m[i]<=60)
                p[i]=6;
            else if (m[i]>=40 && m[i]<=50)
                p[i]=5;
            else
                p[i]=0;
        }
        void calculate () {
            for (int i=0; i<8; i++)
            {
                gt+=c[i]*p[i];
            }
            for (int i=0; i<8; i++)
            {
                sum+=c[i];
            }
            sgpa=gt/sum;
        }
        void display () {
            System.out.println("Name : " + name);
            System.out.println("USN : " + usn);
            System.out.println("SGPA : " + sgpa);
        }
    }
}

```

```

class student {
    public static void main(String args) {
        Stud_det s1[] = new Stud_det(5);
        for (int i=0; i<5; i++)
        {
            System.out.println("Enter details of student " + (i+1));
            s1[i].getdetails();
        }
        for (int i=0; i<5; i++)
        {
            s1[i].gradepoint();
            s1[i].calculate();
        }
        for (int i=0; i<5; i++)
        {
            System.out.println("Student - " + (i+1));
            s1[i].display();
        }
    }
}

Output :
Enter details of student 1:
Enter name
Anu
Enter USN:
IBM23CS004
Enter marks of subject 1
90
Enter credits for subject 1
10
Enter marks of subject 2
85
Enter credits of subject 2
9
Enter marks of subject 3
77
Enter credits of subject 3
8
Enter marks of subject 4
75
Enter credits of subject 4
7
Enter marks of subject 5
76
Enter credits of subject 5
8
Enter marks of subject 6
88
Enter credits of subject 6
9
Enter marks of subject 7
56
Enter credits of subject 7
6
Enter marks of subject 8
89
Enter credits of subject 8
8
o/p Screen
Student -1
Name Anu
USN IBM23CS004
SGPA = 8.0
16/10/24

```

## CODE:

```

import java.util.*;

class Stud_det {
    int m[] = new int[8];
    int c[] = new int[8];
    int p[] = new int[8];
    int g, sum;
    String name, usn;
    double sgpa;
    Scanner s = new Scanner(System.in);

    void getdetails() {
        System.out.println("Enter name:");
        name = s.nextLine();
        System.out.println("Enter usn:");
        usn = s.nextLine();
        for (int i = 0; i < 8; i++) {
            System.out.println("Enter marks of subject:" + (i+1));
            m[i] = s.nextInt();
            System.out.println("Enter credits for subject:" + (i+1));
            c[i] = s.nextInt();
        }
    }

    void gradepoint() {
        for (int i = 0; i < 8; i++) {
            if (m[i] >= 90 && m[i] <= 100)
                p[i] = 10;
            else if (m[i] >= 80 && m[i] < 90)
                p[i] = 9;
            else if (m[i] >= 70 && m[i] < 80)
                p[i] = 8;
            else if (m[i] >= 60 && m[i] < 70)
                p[i] = 7;
            else if (m[i] >= 50 && m[i] < 60)
                p[i] = 6;
            else if (m[i] >= 40 && m[i] < 50)
                p[i] = 5;
            else
                p[i] = 0;
        }
    }

    void calculate() {
        for (int i = 0; i < 8; i++) {
            g += c[i] * p[i];
        }
        sum += c[i];
    }

    void display() {
        System.out.println("Name: " + name);
        System.out.println("USN: " + usn);
        System.out.println("SGPA: " + sgpa);
    }
}

class student1 {
    public static void main(String a[]) {
        Stud_det s1[] = new Stud_det[3];
        for (int i = 0; i < 3; i++) {
            s1[i] = new Stud_det();
        }
        for (int i = 0; i < 3; i++) {
            System.out.println("Enter details of student:" + (i+1));
            s1[i].getdetails();
        }
        for (int i = 0; i < 3; i++) {
            s1[i].gradepoint();
            s1[i].calculate();
        }
        for (int i = 0; i < 3; i++) {
            System.out.println("Student-" + (i+1));
            s1[i].display();
        }
    }
}

```

```
C:\Users\bmsce>cd Documents
C:\Users\bmsce\Documents>cd 1BM23CS050
C:\Users\bmsce\Documents\1BM23CS050>javac student1.java
C:\Users\bmsce\Documents\1BM23CS050>java student1
Enter details of student:1
Enter name:
Avi
Enter usn:
1BM23CS004
Enter marks of subject:1
90
Enter credits for subject:1
10
Enter marks of subject:2
88
Enter credits for subject:2
8
Enter marks of subject:3
77
Enter credits for subject:3
8
Enter marks of subject:4
75
Enter credits for subject:4
8
Enter marks of subject:5
76
Enter credits for subject:5
8
Enter marks of subject:6
88
Enter credits for subject:6
9
Enter marks of subject:7
56
Enter credits for subject:7
6
Enter marks of subject:8
89
Enter credits for subject:8
8
```

```
Enter details of student:2
Enter name:
Mridula
Enter usn:
1BM23CS001
Enter marks of subject:1
98
Enter credits for subject:1
10
Enter marks of subject:2
99
Enter credits for subject:2
10
Enter marks of subject:3
88
Enter credits for subject:3
9
Enter marks of subject:4
79
Enter credits for subject:4
8
Enter marks of subject:5
86
Enter credits for subject:5
9
Enter marks of subject:6
67
Enter credits for subject:6
7
Enter marks of subject:7
45
Enter credits for subject:7
5
Enter marks of subject:8
54
Enter credits for subject:8
6
Enter details of student:3
Enter name:
Xavier
Enter usn:
1BM23CS090
Enter marks of subject:1
10
Enter credits for subject:1
0
Enter marks of subject:2
57
Enter credits for subject:2
6
Enter marks of subject:3
76
Enter credits for subject:3
7
Enter marks of subject:4
88
Enter credits for subject:4
8
Enter marks of subject:5
45
Enter credits for subject:5
5
Enter marks of subject:6
34
Enter credits for subject:6
0
Enter marks of subject:7
54
Enter credits for subject:7
5
Enter marks of subject:8
66
Enter credits for subject:8
6
Enter details of student:4
Enter name:
Student-1
Name:Avi
USN:1BM23CS004
SGPA=:8.0
Student-2
Name:Mridula
USN:1BM23CS001
SGPA=:8.0
Student-3
Name:Xavier
USN:1BM23CS090
SGPA=:7.0
```

```
Enter details of student:3
Enter name:
Xavier
Enter usn:
1BM23CS090
Enter marks of subject:1
10
Enter credits for subject:1
0
Enter marks of subject:2
57
Enter credits for subject:2
6
Enter marks of subject:3
76
Enter credits for subject:3
7
Enter marks of subject:4
88
Enter credits for subject:4
8
Enter marks of subject:5
45
Enter credits for subject:5
5
Enter marks of subject:6
34
Enter credits for subject:6
0
Enter marks of subject:7
54
Enter credits for subject:7
5
Enter marks of subject:8
66
Enter credits for subject:8
6
Enter details of student:8
Enter name:
Student-1
Name:Avi
USN:1BM23CS004
SGPA=:8.0
Student-2
Name:Mridula
USN:1BM23CS001
SGPA=:8.0
Student-3
Name:Xavier
USN:1BM23CS090
SGPA=:7.0
C:\Users\bmsce\Documents\1BM23CS0
```

## 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.

### ALGORITHM:

```

lab-03
Create a class which contains four members : name,
author , price, num-pages include

import java.util.Scanner;
class Book {
    private String name;
    private String author;
    private int price;
    private int numPages;
    String name, int, author, price, numPages) {
        this.name = name;
        this.author = author;
        this.price = price;
        this.numPages = numPages;
    }
    public void setName(String name) {
        this.name = name;
    }
    public void setAuthor(String author) {
        this.author = author;
    }
    public void setPrice(int price) {
        this.price = price;
    }
    public void setNumPages(int no) {
        this.numPages = no;
    }
}

public String getName() {
    return name;
}
public String getAuthor() {
    return author;
}
public int getPrice() {
    return price;
}
public int getNumPages() {
    return numPages;
}
public String toString() {
    return ("Books: " + name + "\n" + "Author: " + author + "\n"
           + "Price: " + price + "\n" + "No of pages: " + numPages
           + "\n");
}
}

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

```

```

2
System.out.println("Enter price");
int price = sc.nextInt();
System.out.println("Enter no of pages : ");
int numPages = sc.nextInt();
books[i] = new Book(name, author, price, numPages);
}

for (int i = 0; i < n; i++) {
}
System.out.println(books[i].toString());
}
}

Output -
Enter no of books : 2
BOOK : MATH
AUTHOR NAME : RD SHARMA
PRICE : 500
NO OF PAGES : 350
MATH
Enter the author name :
RD SHARMA
Enter the price :
PRICE : 450
NO OF PAGES : 500
350
Enter the name : SCIENCE
SCIENCE
Enter the author name :
GOYAL BROTHERS
Enter the price :
PRICE : 500
NO OF PAGES : 500
0

```

## CODE:

```
import java.util.Scanner;
class Book{
    private String name;
    private String author;
    private int price;
    private int num_pages;
    public Book(String name, String author, int price, int num_price){
        this.name=name;
        this.author=author;
        this.price=price;
        this.num_pages=num_pages;
    }
    public void setName(String name){
        this.name=name;
    }
    public void setAuthor(String author){
        this.author=author;
    }
    public void setPrice(int price){
        this.price=price;
    }
    public void setNumPages(int no){
        this.num_pages=no;
    }
    public String getName(){
        return name;
    }
    public String getAuthor(){
        return author;
    }
    public int getPrice(){
        return price;
    }
    public int getNumPages(){
        return num_pages;
    }
    public String toString(){
        return("BOOK: "+name+"\n"+ "AUTHOR: "+author+"\n"+ "PRICE: "+price+
    }
}
```

```
Enter no.of books:
2
Enter the name:
Enter the author name:
MATH RD SHARMA
Enter the price:
560
Enter the no.of pages:
350
Enter the name:
Enter the author name:
SCIENCE GOYAL BROTHERS
Enter the price:
450
Enter the no.of pages:
500
BOOK:
AUTHOR: MATH RD SHARMA
PRICE: 560
No.of PAGES: 0

BOOK:
AUTHOR: SCIENCE GOYAL BROTHERS
PRICE: 450
No.of PAGES: 0

C:\Users\bmsce\Documents\Archita>s
```

```
    return("BOOK: "+name+"\n"+ "AUTHOR: "+author+"\n"+ "PRICE: "+price+
}
}
public class Test{
public static void main(String [] args){
    Scanner sc=new Scanner(System.in);

    System.out.println("Enter no.of books: ");
    int n = sc.nextInt();
    Book[] books=new Book[n];
    for(int i=0;i<n;i++)
    {
        System.out.println("Enter the name: ");
        String name=sc.nextLine();
        System.out.println("Enter the author name: ");
        String author=sc.nextLine();
        System.out.println("Enter the price: ");
        int price=sc.nextInt();
        System.out.println("Enter the no.of pages: ");
        int num_pages=sc.nextInt();
        books[i]=new Book(name,author,price,num_pages);
    }
    for(int i=0;i<n;i++)
    {
        System.out.println(books[i].toString());
    }
}
};
```

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

### ALGORITHM:

```

23/10/14
Lab No. 4
Develop a Java program to create an abstract class
named Shape that contains two integers and an
empty method.

import java.util.Scanner;
abstract class Shape {
    int dimension1;
    int dimension2;
}
public Shape(int dimension1, int dimension2) {
    this.dimension1 = dimension1;
    this.dimension2 = dimension2;
}
public abstract void printArea();
}

class Rectangle extends Shape {
    public Rectangle(int length, int width) {
        dimension1 = length;
        dimension2 = width;
    }
    public void printArea() {
        int area = dimension1 * dimension2;
        System.out.println("Area of rectangle: " + area);
    }
}

class Triangle extends Shape {
    public Triangle(int base, int height) {
        dimension1 = base;
        dimension2 = height;
    }
    public void printArea() {
        double area = 0.5 * dimension1 * dimension2;
        System.out.println("Area of triangle: " + area);
    }
}

```

```

public void printArea() {
    double area = 0.5 * dimension1 * dimension2;
    System.out.println("Area of triangle: " + area);
}

class Circle extends Shape {
    public Circle(int radius) {
        dimension1 = radius;
        dimension2 = 0;
    }
    public void printArea() {
        double area = Math.PI * dimension1 * dimension2;
        System.out.println("Area of circle: " + area);
    }
}

public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter l & w of □: ");
        int length = scanner.nextInt();
        int width = scanner.nextInt();
        Shape rectangle = new Rectangle(length, width);
        rectangle.printArea();

        scanner.close();
    }
}

Output -
Enter length and width for rectangle:
2 3
Area of rectangle: 6
Enter base & height for triangle:
2 4
Area of triangle: 4.0
Enter radius for circle:
3
Area of circle: 28.274335
OP seen
Solved

```

## CODE:

```
import java.util.Scanner;

abstract class Shape {
    int dimension1;
    int dimension2;

    public Shape() {
        this.dimension1 = 0;
        this.dimension2 = 0;
    }

    public Shape(int dimension1, int dimension2) {
        this.dimension1 = dimension1;
        this.dimension2 = dimension2;
    }

    public abstract void printArea();
}

class Rectangle extends Shape {
    public Rectangle(int length, int width) {
        dimension1 = length;
        dimension2 = width;
    }

    public void printArea() {
        int area = dimension1 * dimension2;
        System.out.println("Area of Rectangle: " + area);
    }
}

class Triangle extends Shape {
    public Triangle(int base, int height) {
        dimension1 = base;
        dimension2 = height;
    }

    public void printArea() {
        double area = 0.5 * dimension1 * dimension2;
        System.out.println("Area of Triangle: " + area);
    }
}

public void printArea() {
    double area = 0.5 * dimension1 * dimension2;
    System.out.println("Area of Circle: " + area);
}

class Circle extends Shape {
    public Circle(int radius) {
        dimension1 = radius;
        dimension2 = 0;
    }

    public void printArea() {
        double area = Math.PI * dimension1 * dimension1;
        System.out.println("Area of Circle: " + area);
    }
}

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

        System.out.println("Enter length and width for Rectangle:");
        int length = scanner.nextInt();
        int width = scanner.nextInt();
        Shape rectangle = new Rectangle(length, width);
        rectangle.printArea();

        System.out.println("Enter base and height for Triangle:");
        int base = scanner.nextInt();
        int height = scanner.nextInt();
        Shape triangle = new Triangle(base, height);
        triangle.printArea();

        System.out.println("Enter radius for Circle:");
        int radius = scanner.nextInt();
        Shape circle = new Circle(radius);
        circle.printArea();

        scanner.close();
    }
}
```

```
C:\Users\bmsce>cd Documents
C:\Users\bmsce\Documents>cd Archita
C:\Users\bmsce\Documents\Archita>javac main.java
C:\Users\bmsce\Documents\Archita>java main
Enter length and width for Rectangle:
2 3
Area of Rectangle: 6
Enter base and height for Triangle:
2 4
Area of Triangle: 4.0
Enter radius for Circle:
3
Area of Circle: 28.274333882308138
C:\Users\bmsce\Documents\Archita>
```

## LAB 5

Develop a Java program to create a class of Bank that maintains two kinds of accounts for its customers, one called a 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.

### ALGORITHM:

Lab - 05

Develop a Java program to create a class Bank that takes 2 kinds of accounts for its customers.

```

import java.util.Scanner;
class Account {
    private String customerName;
    private int accNo;
    protected double balance;
    public Account (String customerName, int accNo,
                    double balance) {
        this.customerName = customerName;
        this.accNo = accNo;
        this.balance = balance;
    }
    public double getBalance () {
        return balance;
    }
    public void deposit (double amt) {
        if (amt > 0)
            balance += amt;
        System.out.println ("Deposited : " + amt);
    }
    else
        System.out.println ("Deposited amount must be positive");
    }
}

```

public void withdraw (double amt) {
 if (amt < getBalance ())
 balance -= amt;
 System.out.println ("Withdrawal : " + amt + " balance is " + balance);
}
else
 System.out.println ("Insufficient funds");
}

public void displayBalance () {
 System.out.println ("Current Balance : " + balance);
}

class SavingsAccount extends Account {
 private double interestRate;
 public SavingsAccount (String customerName, int accNumber,
 double initialBalance, double interestRate) {
 super (customerName, accNumber, initialBalance);
 this.interestRate = interestRate;
 }
 public void computeAndDeposit () {
 double interest = getBalance () \* interestRate / 100;
 deposit (interest);
 }
}

```

class CurrentAc extends Account {
    private double balance;
    private double minBal;

    public CurrentAc (String custName, int accNumber) {
        super (accNumber, accNumber, minBal);
        balance = minBal;
        minBal = minimumBal;
        serviceChg = serviceChg;
    }

    void checkMinBalance () {
        if (getBalance () < minBal)
            System.out.println ("Balance is below minimum.");
        else
            System.out.println ("Deducted service charge : " + serviceChg);
        System.out.println ("Balance after deduction is : " + balance);
    }

    public class Bank {
        public static void main (String [] args) {
            Scanner sc = new Scanner (System.in);
            System.out.println ("Enter customer name : ");
            String name = sc.nextLine ();
            System.out.println ("Enter account number : ");
            int acc_no = sc.nextInt ();
        }
    }
}

```

```

class
    System.out.println("Enter account number");
    int accountNo = input.nextInt();
    System.out.println("Enter initial balance");
    double initialBalance = input.nextDouble();
    SavingsAccount account = new SavingsAccount(accountNo, initialBalance);

    do
        System.out.println("1. Deposit");
        System.out.println("2. Withdraw");
        System.out.println("3. Display Balance");
        System.out.println("4. Exit");
        int choice = input.nextInt();

        if (choice == 1) {
            System.out.println("Enter amount to be deposited");
            double amt = input.nextDouble();
            account.deposit(amt);
        } else if (choice == 2) {
            System.out.println("Enter amount to be withdrawn");
            double amt = account.nextDouble();
            account.withdraw(amt);
        } else if (choice == 3) {
            account.displayBalance();
        } else if (choice == 4) {
            System.out.println("Thank you for using our services");
            System.out.println("Have a nice day!");
            break;
        }
    } while (true);
}

```

Signature: [Signature]

## CODE:

```

import java.util.Scanner;

class Account {
    private String customer_name;
    private int acc_no;
    protected double balance;

    public Account(String customer_name, int acc_no, double balance) {
        this.customer_name = customer_name;
        this.acc_no = acc_no;
        this.balance = balance;
    }

    public double getBalance() {
        return balance;
    }

    public void deposit(double amount) {
        if (amount > 0) {
            balance += amount;
            System.out.println("Deposited: " + amount);
        } else {
            System.out.println("Deposit amount must be positive.");
        }
    }

    public void withdraw(double amount) {
        if(amount>getBalance()){
            balance-=amount;
            System.out.println("withdraw:"+amount+" balance is:"+balance);
        }
        else
            System.out.println("Insufficient funds!!");
    }

    public void displayBalance(){
        System.out.println("Current Balance: " + balance);
    }
}

class SavingsAccount extends Account {
    private double interestRate;

    public SavingsAccount(String customerName, int accountNumber, double initialBalance, double interestRate) {
        super(customerName, accountNumber, initialBalance);
        this.interestRate = interestRate;
    }

    public void computeAndDepositInterest() {
        double interest = getBalance() * interestRate / 100;
        deposit(interest);
    }
}

class CurrentAccount extends Account {
    private double minimumBalance;
    private double serviceCharge;

    public CurrentAccount(String customerName, int accountNumber, double initialBalance, double minimumBalance, double serviceCharge) {
        super(customerName, accountNumber, initialBalance);
        this.minimumBalance = minimumBalance;
        this.serviceCharge = serviceCharge;
    }

    public void checkMinimumBalance() {
        if (getBalance() < minimumBalance) {
            System.out.println("Balance is below minimum");
            balance-=serviceCharge;
            System.out.println("Deducted service charge:" + serviceCharge);
            System.out.println("Balance after deduction is:" + balance);
        }
    }
}

public class Bank {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("enter customer name:");
        String name=sc.nextLine();
        System.out.println("enter accno:");
        int acc_no=sc.nextInt();
        System.out.println("enter initial balance:");
        double balance=sc.nextDouble();
        System.out.println("enter minimum balance:");
        double minimum_balance=sc.nextDouble();
        System.out.println("enter interest rate:");
        double interest_rate=sc.nextDouble();
        System.out.println("enter service charge:");
        double service_charge=sc.nextDouble();
        System.out.println("Enter choice:\n 1.Current acc\n 2.Savings acc");
        int ch=sc.nextInt();
        System.out.println("Customer name is:" + name + "\nAccount number:" + acc_no + "\n ARCHITA V-1BM23CS050");

        switch(ch){
            case(1):
                System.out.println("enter interest rate:");
                double interest_rate=sc.nextDouble();
                System.out.println("enter service charge:");
                double service_charge=sc.nextDouble();
                System.out.println("Enter choice:\n 1.Current acc\n 2.Savings acc");
                int ch1=sc.nextInt();
                if(ch1==1){
                    System.out.println("enter amount to be deposited:");
                    double amt=sc.nextDouble();
                    ca.deposit(amt);
                }
                else if(ch1==2){
                    System.out.println("enter amount to withdraw:");
                    double amt=sc.nextDouble();
                    ca.withdraw(amt);
                }
                else if(ch1==3){
                    ca.displayBalance();
                }
                else{
                    System.exit(0);
                }
                jwhile(true);
            }

            case(2):
                System.out.println("account is savings type");
                SavingsAccount sa=new SavingsAccount(name,acc_no,balance,interest_rate);
                do{ System.out.println("enter choice:\n 1.deposit\n 2.withdraw\n 3.display balance");
                int c1=sc.nextInt();
                if(c1==1){
                    System.out.println("enter amount to be deposited:");
                    double amt=sc.nextDouble();
                    sa.deposit(amt);
                }
                else if(c1==2){
                    System.out.println("enter amount to withdraw:");
                    double amt=sc.nextDouble();
                    sa.withdraw(amt);
                }
                else if(c1==3){
                    sa.computeAndDepositInterest();
                    sa.displayBalance();
                }
                else{
                    System.exit(0);
                }
                }while(true);
        }
    }
}

```

C:\Users\Admin\Desktop\1BM23CS050>java Bank  
enter customer name:  
John  
enter accno:  
10909  
enter initial balance:  
10000  
enter minimum balance:  
3  
enter interest rate:  
54  
enter service charge:  
54  
Enter choice:  
1.Current acc  
2.Savings acc  
1  
Customer name is:John  
Account number:1  
ARCHITA V-1BM23CS050  
account is current type  
enter choice:  
1.deposit  
2.withdraw  
3.display balance  
1  
enter amount to be deposited:  
1200  
Deposited: 1200.0  
enter choice:  
1.deposit  
2.withdraw  
3.display balance  
2  
enter amount to withdraw:  
2390  
Withdraw:2390.0 balance is:9719.0  
enter choice:  
1.deposit  
2.withdraw  
3.display balance  
3  
Balance is below minimum  
Deducted service charge:54.0  
Balance after deduction is:9665.0  
Current Balance: 9665.0  
enter choice:  
1.deposit  
2.withdraw  
3.display balance

## LAB 6

Create a package CIE which has two classes - Personal 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 Personal. 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.

### ALGORITHM:

```

Lab 6
make a package CIE which has 2 classes - student , internals

package com.CIE;
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;
}
public void displayDetails()
{
    System.out.println("USN: " + usn);
    System.out.println("Name: " + name);
    System.out.println("Semester: " + sem);
}
public class Internals
{
    public int[] internalMarks;
    public Internals (String name, String usn, int sem, int[] marks)
    {
        super(name, usn, sem);
        if(marks.length != 5)
        {
            System.out.println("Enter 5 marks");
        }
    }
}

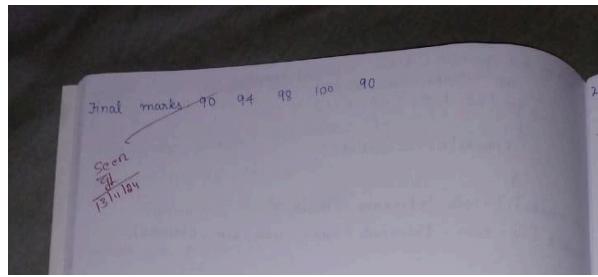
package com.SEE;
import com.CIE.Student;
public class External extends Student
{
    public int[] externalMarks;
    public External (String name, String usn, int sem, int[] marks)
    {
        super(name, usn, sem);
        if(marks.length != 5)
        {
            System.out.println("Enter 5 subjects");
            this.externalMarks = marks;
        }
    }
}

public void displayMarks()
{
    System.out.println("SEE Marks");
    for(int i=0; i<5; i++)
    {
        System.out.println(marks[i]);
    }
}

import com.CIE.*;
import SEE.*;
import java.util.Scanner;
public class Main
{
    public static void main (String [] args)
    {
        Scanner sc = new Scanner (System.in);
        System.out.println("Enter no of students: ");
        int n = sc.nextInt();
        Internals [] students = new Internals [n];
        Internals [] intMarks = new Internals [n];
        for(int i=0; i<n; i++)
        {
            System.out.println("Enter usn: ");
            String usn = sc.nextLine();
            System.out.println("Enter name: ");
            String name = sc.nextLine();
            System.out.println("Enter semester: ");
            int sem = sc.nextInt();
            System.out.println("Enter internal marks: ");
            int [] IMarks = new int [5];
            for(int j=0; j<5; j++)
            {
                IMarks[j] = sc.nextInt();
            }
        }
    }
}

System.out.println("Enter external marks: ");
int marks [] = new int [5];
for(int k=0; k<5; k++)
{
    marks[k] = sc.nextInt();
}
int finalMarks [] = new Internals [n];
for(int l=0; l<n; l++)
{
    finalMarks[l] = new External (name, usn, sem, marks);
}
System.out.println("Final marks: ");
for(int m=0; m<n; m++)
{
    System.out.println(finalMarks[m].externalMarks);
}
System.out.println("Final: ");
for(int o=0; o<n; o++)
{
    System.out.println(finalMarks[o].finalMarks);
}
System.out.println("Output");
System.out.println("Number of students: 1");
System.out.println("usn: 123456789");
System.out.println("name: ABC");
System.out.println("Semester: 3");
System.out.println("Internal marks: 40 39 36 34 35");
System.out.println("External marks: 50 55 60 67 80");
System.out.println("Final marks: 50 55 64 67 80");

```



## CODE:

```

package CIE;
import java.util.Scanner;
public class Internals extends Student{
protected int marks[]={new int[5];
public void inputCIEmarks()
{
Scanner sc=new Scanner(System.in);
System.out.println("Enter CIE marks:");
for(int i=0;i<5;i++)
{
System.out.println("Enter sub"+(i+1)+"marks:");
marks[i]=sc.nextInt();
}
}

package CIE;
import java.util.Scanner;
public class Student{
protected String usn=new String();
protected String name=new String();
protected int sem;
public void inputStudentDetails()
{
Scanner sc=new Scanner(System.in);
System.out.println("Enter usn,name and present semester :");
usn=sc.next();
name=sc.next();
sem=sc.nextInt();
}
public void displayStudentDetails()
{
System.out.println("Student details:
\nUsn:"+usn+"\nName:"+name+"\nPresent Sem:"+sem);
}
}

System.out.println("Enter SEE marks(out of 50):");
for(int i=0;i<5;i++)
{
System.out.println("Enter sub"+(i+1)+"marks:");
marks[i]=sc.nextInt();
}

public void calcFinalMarks()
{
for(int i=0;i<5;i++)
{
finalMarks[i]=super.marks[i]+marks[i];
}
}

public void displayFinalMarks()
{
System.out.println("Final Marks:");
for(int i=0;i<5;i++)
{
System.out.print("Sub"+(i+1)+":"+finalMarks[i]+" ");
}
}

System.out.println();
}

public void displayStudentDetails()
{
System.out.println("Student details:
\nUsn:"+usn+"\nName:"+name+"\nPresent Sem:"+sem);
displayFinalMarks();
}
}

package SEE;
import CIE.*;
import java.util.Scanner;

public class Externals extends Internals{
protected int marks[];
protected int finalMarks[];
Scanner sc=new Scanner(System.in);

public Externals(){
marks=new int[5];
finalMarks=new int[5];
}

public void inputSEEmarks() {

System.out.println("Enter SEE marks(out of 50):");
for(int i=0;i<5;i++)
{
System.out.println("Enter sub"+(i+1)+"marks:");
marks[i]=sc.nextInt();
}
}

public void calcFinalMarks()
{
for(int i=0;i<5;i++)
{
finalMarks[i]=super.marks[i]+marks[i];
}
}

public void displayFinalMarks()
{
}
}

```

```
Enter number of Students:  
2  
Enter Student1 details:  
Enter usn,name and present semester :  
1  
sai  
3  
Enter CIE marks:  
Enter sub1marks:  
45  
Enter sub2marks:  
50  
Enter sub3marks:  
45  
Enter sub4marks:  
43  
Enter sub5marks:  
50  
Enter SEE marks(out of 50):  
Enter sub1marks:  
43  
Enter sub2marks:  
46  
Enter sub3marks:  
42  
Enter sub4marks:  
32  
Enter sub5marks:  
12  
Student details:  
Usn:1  
Name:sai  
Present Sem:3  
Final Marks:  
Sub1:88 Sub2:96 Sub3:87 Sub4:75 Sub5:62
```

## LAB 7

Write a program that demonstrates handling of exceptions in inheritance trees. Create a base class called “Father” and a 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 uses both father and son’s age and throws an exception if son’s age is >=father’s age.

### ALGORITHM:

20/11/24  
Lab -07

NBT that demonstrates handling of exception in inheritance tree. Create a base class “father” & derived class “son” which extends base class. In father class, implement a constructor which takes the age and throws the exception WrongAge( ) when input age <0. In son class, implement a constructor that uses both father & son’s age & throws an exception if son’s age >= father’s age.

```

import java.util.Scanner;
class WrongAgeException extends Exception
{
    public WrongAgeException (String message)
    {
        super (message);
    }
}
class Father
{
    int fatherage;
    public Father (int age) throws WrongAgeException
    {
        if (age < 0)
        {
            throw new WrongAgeException ("Age cannot be negative
for father");
        }
        this.fatherage = age;
    }
    System.out.println ("Father's age is set to : " + age);
}
    
```

class son extends Father
{
 int sonage;
 public Son (int fatherage, int sonage) throws
 WrongAgeException
 {
 super (fatherage);
 if (sonage < 0)
 {
 throw new WrongAgeException ("Age cannot be negative");
 }
 if (sonage > fatherage)
 {
 throw new WrongAgeException ("Son's age cannot be
greater than or equal to father's age");
 }
 this.sonage = sonage;
 }
 System.out.println ("Son's age is set to : " + sonage);
}
public class FatherSon
{
 public static void main (String [] args)
 {
 Scanner scanner = new Scanner (System.in);
 try
 {
 System.out.println ("Enter Father's age : ");
 int fatherage = scanner.nextInt ();
 Father father = new Father (fatherage);

System.out.println ("Enter Son's age : ");
 int sonage = scanner.nextInt ();
 Son son = new Son (fatherage, sonage);
 catch (WrongAgeException e)
 {
 System.out.println ("Exception : " + e.getMessage ());
 }
 catch (WrongAgeException e)
 {
 System.out.println ("Uncaught exception : " + e.getMessage ());
 }
 finally
 {
 System.out.println ("Execution completed.");
 scanner.close ();
 }
 }
}

Output :

Enter Father's age : -10  
Exception : Age cannot be negative for Father  
Execution completed

Enter Father's age = 55  
Father's age is set to 55

Enter Son's age = 23  
Father's age is set to 65

Son's age is set to 23  
Execution completed

Enter Father's age = 44  
Father's age is set to 44

Enter Son's age = -10  
Father's age is set to 44

Exception Age cannot be negative for Son

Execution completed

✓ H.A.

## CODE:

```

import java.util.Scanner;

class WrongAgeException extends Exception {
    public WrongAgeException(String message) {
        super(message);
    }
}

class Father {
    int fatherAge;

    public Father(int age) throws WrongAgeException {
        if (age < 0) {
            throw new WrongAgeException("Age cannot be negative for Father.");
        }
        this.fatherAge = age;
        System.out.println("Father's age is set to: " + age);
    }
}

class Son extends Father {
    int sonAge;

    public Son(int fatherAge, int sonAge) throws WrongAgeException {
        super(fatherAge);
        if (sonAge < 0) {
            throw new WrongAgeException("Age cannot be negative for Son.");
        }
        if (sonAge >= fatherAge) {
            throw new WrongAgeException("Son's age cannot be greater than or equal to Father's age.");
        }
        this.sonAge = sonAge;
        System.out.println("Son's age is set to: " + sonAge);
    }
}

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

        try {
            System.out.print("Enter Father's age: ");
            int fatherAge = scanner.nextInt();

            Father father = new Father(fatherAge);

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

```

```

        }
        this.fatherAge = age;
        System.out.println("Father's age is set to: " + age);
    }

    class Son extends Father {
        int sonAge;
        public Son(int fatherAge, int sonAge) throws WrongAgeException {
            super(fatherAge);
            if (sonAge < 0) {
                throw new WrongAgeException("Age cannot be negative for Son.");
            }
            if (sonAge >= fatherAge) {
                throw new WrongAgeException("Son's age cannot be greater than or equal to Father's age.");
            }
            this.sonAge = sonAge;
            System.out.println("Son's age is set to: " + sonAge);
        }
    }

    public class FatherSon {
        public static void main(String[] args) {
            Scanner scanner = new Scanner(System.in);
            try {
                System.out.print("Enter Father's age: ");
                int fatherAge = scanner.nextInt();

                Father father = new Father(fatherAge);

                System.out.print("Enter Son's age: ");
                int sonAge = scanner.nextInt();

                Son son = new Son(fatherAge, sonAge);
            } catch (WrongAgeException e) {
                System.out.println("Exception: " + e.getMessage());
            } catch (Exception e) {
                System.out.println("Unexpected Exception: " + e.getMessage());
            } finally {
                System.out.println("Execution completed.");
                scanner.close();
            }
        }
    }
}

```

```

C:\Users\Admin>cd desktop

C:\Users\Admin\Desktop>javac FatherSon.java

C:\Users\Admin\Desktop>java FatherSon
Enter Father's age: -10
Exception: Age cannot be negative for Father.
Execution completed.

C:\Users\Admin\Desktop>java FatherSon
Enter Father's age: 55
Father's age is set to: 55
Enter Son's age: 23
Father's age is set to: 55
Son's age is set to: 23
Execution completed.

C:\Users\Admin\Desktop>java FatherSon
Enter Father's age: 44
Father's age is set to: 44
Enter Son's age: -10
Father's age is set to: 44
Exception: Age cannot be negative for Son.
Execution completed.

```

## LAB 8

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.

### ALGORITHM:

27/11/24  
Lab - 08

Write a program which creates 2 threads, one thread displaying “BMS College of Engineering” once every 10 sec & other displaying “CSE” every 2 seconds

```
class BMSCollegeThreads extends Thread
{
    private volatile boolean running = true; // private variable
    public void stopRunning() { // public method
        running = false;
    }
    public void run() { // run method
        try {
            while (running) { // while loop
                System.out.println("BMS College of Engineering");
                Thread.sleep(10000); // sleep for 10 sec
            }
        } catch (InterruptedException e) {
            System.out.println("BMSCollegeThread interrupted");
        }
        System.out.println("BMSCollegeThread stopped");
    }
}
class CSEThreads extends Thread
{
    private volatile boolean running = true; // private variable
}
```

```

public void stopRunning()
{
    running = false;
}

public void run()
{
try
{
    while (running)
    {
        System.out.println("CSE");
        Thread.sleep(2000);
    }
}
catch (InterruptedException e)
{
    System.out.println("CSE thread interrupted.");
}
System.out.println("CSE thread stopped.");
}

public class MultiThreadDisplay {
public static void main (String [] args)
{
    BMSCollegeThread thread1 = new BMSCollegeThread();
    CSEThread thread2 = new CSEThread();

    thread1.start();
    thread2.start();

    try
    {
        Thread.sleep(3000);
    }
    catch (InterruptedException e)
    {
        System.out.println("Main thread interrupted.");
    }
    thread1.stopRunning();
    thread2.stopRunning();
}
}

Output:
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS college of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College Thread stopped
CSE Thread stopped
Main Thread exiting

```

## CODE:

```

class BMSCollegeThread extends Thread {
    private volatile boolean running = true;

    public void stopRunning() {
        running = false;
    }

    public void run() {
        try {
            while (running) {
                System.out.println("BMS College of Engineering");
                Thread.sleep(10000);
            }
        } catch (InterruptedException e) {
            System.out.println("BMSCollegeThread interrupted.");
        }
        System.out.println("BMSCollegeThread stopped.");
    }
}

class CSEThread extends Thread {
    private volatile boolean running = true;

    public void stopRunning() {
        running = false;
    }

    public void run() {
        try {
            while (running) {
                System.out.println("CSE");
                Thread.sleep(2000);
            }
        } catch (InterruptedException e) {
            System.out.println("CSEThread interrupted.");
        }
        System.out.println("CSEThread stopped.");
    }
}

public class MultiThreadDisplay {
    public static void main(String[] args) {
        BMSCollegeThread thread1 = new BMSCollegeThread();
        CSEThread thread2 = new CSEThread();

        thread1.start();
        thread2.start();

        try {
            Thread.sleep(30000);
        } catch (InterruptedException e) {
            System.out.println("Main thread interrupted.");
        }

        thread1.stopRunning();
        thread2.stopRunning();

        try {
            thread1.join();
            thread2.join();
        } catch (InterruptedException e) {
            System.out.println("Main thread interrupted while waiting.");
        }

        System.out.println("Main thread exiting.");
    }
}

```

```

C:\Users\Admin\Desktop>javac MultiThreadDisplay.java

C:\Users\Admin\Desktop>java MultiThreadDisplay
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMSCollegeThread stopped.
CSEThread stopped.
Main thread exiting.

C:\Users\Admin\Desktop>

```

## 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

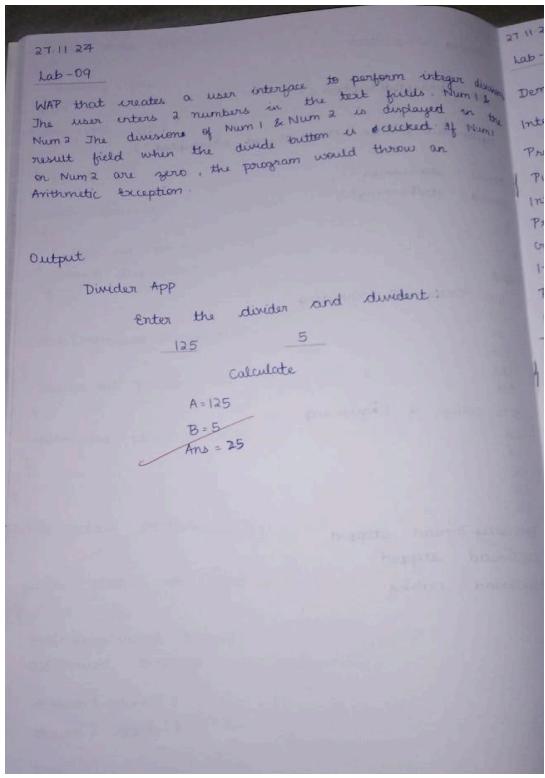
ALGORITHM:

```
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 divider and divident:");
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(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(ArithmetricException e){
alab.setText("");
blab.setText("");
anslab.setText("");
err.setText("B should be NON zero!");
}
}
});
jfrm.setVisible(true);
}

public static void main(String args[]){
SwingUtilities.invokeLater(new Runnable(){
public void run(){
new SwingDemo();
}});}}
```



## LAB 10

Demonstrate Inter process Communication and deadlock.

ALGORITHM [Inter process Communication] :

```
class Q {  
    int n;  
    synchronized int get() {  
        System.out.println("Got: " + n);  
        return n;  
    }  
    synchronized void put(int n) {  
        this.n = n;  
        System.out.println("Put: " + n);  
    }  
}  
  
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();  
            i++;  
        }  
    }  
}  
  
class PC {  
    public static void main(String args[]) {  
        Q q = new Q();  
        new Producer(q);  
        new Consumer(q);  
        System.out.println("Press Control-C to stop.");  
    }  
}
```

```

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");
}
}

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 PCFixed {
    public static void main(String args[]) {
        Q q = new Q();
        new Producer(q);
        new Consumer(q);
        System.out.println("Press Control-C to stop.");
    }
}

```

ALGORITHM [Deadlock] :

```

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("M  
ainThread");  
        Thread t = new Thread(this,  
"RacingThread");  
        t.start();  
        a.foo(b);  
        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();
}

```

