

# VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“JnanaSangama”, Belgaum -590014, Karnataka.



## LAB REPORT

on

## OBJECT ORIENTED JAVA PROGRAMMING

*Submitted by*

**Sinchana Hemanth (1BM23CS330)**

*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-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**" carried out by **Sinchana Hemanth (1BM23CS330)**, 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 during the year 2024-25. The Lab report has been approved as it satisfies the academic requirements in respect of **Object-Oriented Java Programming Lab - (23CS3PCOOJ)** work prescribed for the said degree.

**Dr. Nandhini Vineeth**

Associate Professor,  
Department of CSE,  
BMSCE, Bengaluru

**Dr. Kavitha Sooda**

Professor and Head,  
Department of CSE,  
BMSCE, Bengaluru

## **INDEX**

<b>Sl. No.</b>	<b>Date</b>	<b>Experiment Title</b>	<b>Page No.</b>
1	26/9/24	Quadratic equations	1
2	3/10/24	SGPA calculator	5
3	19/10/24	Book database	13
4	24/10/24	Shapes using abstract classes	19
5	19/11/24	Savings account and Current account	26
6	19/11/24	SEE and CIE score calculator using packages	36
7	21/11/24	Father and son age using exception handling	47
8	29/11/24	Printing a string using Threads	53
9	24/12/24	Open Ended Exercise-1	58
10	24/12/24	Open Ended Exercise-2	65

## LABORATORY PROGRAM - 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.

26/9/24

$d = b^2 - 4ac$   
 $d = 1$

Page No.	
Date	

Program I

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

```
import java.util.Scanner;
class Quadeq
{
    public static void main (String [] args)
    {
        Scanner sc = new Scanner (System.in);
        System.out.println ("Enter coefficient of a:");
        double a = sc.nextDouble();
        System.out.println ("Enter coefficient of b:");
        double b = sc.nextDouble();
        System.out.println ("Enter coefficient of c:");
        double c = sc.nextDouble();
        double disc = b*b - 4*a*c;
        if (disc > 0)
        {
            double root1 = (-b + Math.sqrt(disc))/(2*a);
            double root2 = (-b - Math.sqrt(disc))/(2*a);
            System.out.println ("Equation has two real roots: " + root1 + " " + root2);
        }
        else if (disc == 0)
        {
            double root = -b/2*a;
            System.out.println ("Equation has one real solution " + root);
        }
    }
}
```

else

{

System.out.println ("Equation has no real roots");

}

}

}

}

### Output I

Real roots Enter coefficient of a:

and distinct 2

Enter coefficient of b:

5

Enter coefficient of c:

3

Equation has two real roots: -1.0 -1.5

### Output II

Enter coefficient of a:

1

Enter coefficient of b:

4

Enter coefficient of c:

4

Equation has one real solution: -2.0

### Output III

Enter coefficient of a:

5

Enter coefficient of b:

4

Enter coefficient of c:

4

Equation has no real roots

## **PROGRAM:**

```
import java.util.Scanner;

class Quadeq

{
    public static void main (String[] args)

    {
        Scanner sc = new Scanner (System.in);

        System.out.println("Enter coefficient of a:");

        double a = sc.nextDouble();

        System.out.println("Enter coefficient of b:");

        double b = sc.nextDouble();

        System.out.println("Enter coefficient of c:");

        double c = sc.nextDouble();

        double disc = b*b-4*a*c;

        if(disc>0)

        {
            double root1 = (-b+Math.sqrt(disc))/(2*a);

            double root2 = (-b-Math.sqrt(disc))/(2*a);

            System.out.println("Equation has two real roots: "+root1+" "+root2);

        }

        else if(disc==0)

        {
            double root = -b/(2*a);

            System.out.println("Equation has one real solution: "+root);

        }
    }
}
```

```

else
{
    System.out.println("Equation has no real roots");
}

}

```

## OUTPUT:

```

D:\1BM23CS330>java Quadeq
Enter coefficient of a:
2
Enter coefficient of b:
5
Enter coefficient of c:
3
Equation has two real roots: -1.0 -1.5

D:\1BM23CS330>java Quadeq
Enter coefficient of a:
1
Enter coefficient of b:
4
Enter coefficient of c:
4
Equation has one real solution: -2.0

D:\1BM23CS330>java Quadeq
Enter coefficient of a:
5
Enter coefficient of b:
2
Enter coefficient of c:
4
Equation has no real roots

D:\1BM23CS330>

```

## LABORATORY PROGRAM - 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.

3/10/24

Page No.	
Date	

### Program II

Develop java program to create a class Student with members usn, name, array credits and 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 name;
    String usn;
    int numsub;
    int[] credits;
    int[] marks;

    Student(int num)
    {
        numsub = num;
        credits = new int[numsub];
        marks = new int[numsub];
    }

    void acceptDetails()
    {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter USN: ");
        usn = sc.nextLine();
        System.out.print("Enter name: ");
        name = sc.nextLine();
        System.out.print("Enter credits for each subject: ");
    }
}
```

```

    for (int i=0; i< numsub; i++)
    {
        System.out.println ("Credits for subject" + (i+1) + ": ");
        credits[i] = sc.nextInt();
    }

    System.out.println ("Enter marks for each subject:");
    for (int i=0; i< numsub; i++)
    {
        System.out.println ("Marks for subject" + (i+1) + " : ");
        marks[i] = sc.nextInt();
    }

    void display()
    {
        System.out.println ("Student Details : ");
        System.out.println ("USN: " + usn);
        System.out.println ("Name: " + name);
        System.out.println ("Subject" + (i+1) + " Credits: " + credits[i]);
        System.out.println ("Subject-wise Credits and
                           Marks");
        for (int i=0; i< numsub; i++)
        {
            System.out.println ("Subject " + (i+1) + " Credits = "
                               + credit[i] + ", Marks = " + marks[i]);
        }

        System.out.println ("SGPA: " + calculateSGPA());
    }

    double calculateSGPA()
    {
        int totalCredits = 0;
        int totalGradePoints = 0;
        for (int i=0; i< numsum; i++)
    }

```

Page No.	
Date	

```

    {
        totalCredits += credits[i];
        int gradePoint = calculateGradePoint(marks[i]);
        totalGradePoints += gradePoint * credits[i];
    }
    return (double)totalGradePoints / totalCredits;
}

```

```
int calculateGradePoint(int marks)
```

```

{
    if (marks >= 90)
        return 10;
    else if (marks >= 80)
        return 9;
    else if (marks >= 70)
        return 8;
    else if (marks >= 60)
        return 7;
}
```

```

(i) + ", marks = " + marks[i])
else if (marks >= 50)
    return 6;
else if (marks >= 40)
    return 5;
else
    return 0;
}

```

```
g
```

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

```

    Scanner sc = new Scanner(System.in);
    System.out.print("Enter number of subjects: ");
    int numSub = sc.nextInt();
    Student s = new Student(numSub);
    s.details();
    s.display();
}

```

```
g
```

Output

Enter number of subjects : 5

Enter USN : IBM23CS330

Enter name : Sinchana Hemanth

Enter credits for each subject :

Credits for subject 1: 4

Credits for subject 2: 4

Credits for subject 3: 3

Credits for subject 4: 3

Credits for subject 5: 2

Enter marks for each subject:

Marks for subject 1: 98

Marks for subject 2: 95

Marks for subject 3: 91

Marks for subject 4: 92

Marks for subject 5: 96

Student Details :

USN : IBM23CS330

Name: Sinchana Hemanth

Subject-wise Credits and Marks :

Subject 1 : Credits = 4 , Marks = 98

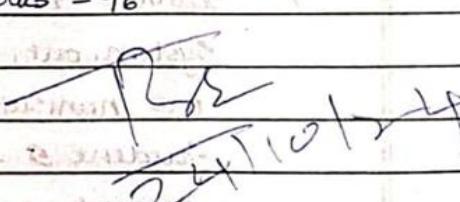
Subject 2 : Credits = 4 , Marks = 95

Subject 3 : Credits = 3 , Marks = 91

Subject 4 : Credits = 3 , Marks = 92

Subject 5 : Credits = 2 , Marks = 96

SGPA : 10.0


  
 24/10/24

## **PROGRAM:**

```
import java.util.Scanner;

class Sgpacalc {

    String usn;

    String name;

    int numSubjects;

    int[] credits;

    int[] marks;

    Sgpacalc(int x) {

        numSubjects = x;

        credits = new int[numSubjects];

        marks = new int[numSubjects];

    }

    void acceptDetails() {

        Scanner sc = new Scanner(System.in);

        System.out.print("Enter USN: ");

        usn = sc.nextLine();

        System.out.print("Enter Name: ");

        name = sc.nextLine();

        System.out.println("Enter credits for each subject:");

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

            System.out.print("Credits for subject " + (i+1) + ": ");

            credits[i] = sc.nextInt();

        }

    }

}
```

```

        System.out.println("Enter marks for each subject:");
        for (int i = 0; i < numSubjects; i++) {
            System.out.print("Marks for subject " + (i+1) + ": ");
            marks[i] = sc.nextInt();
        }
    }

    void displayDetails() {
        System.out.println("\nStudent Details:");
        System.out.println("USN: " + usn);
        System.out.println("Name: " + name);
        System.out.println("\nSubject-wise Credits and Marks:");
        for (int i = 0; i < numSubjects; i++) {
            System.out.println("Subject " + (i+1) + ": Credits = " +
credits[i] + ", Marks = " + marks[i]);
        }
        System.out.println("SGPA: " + calculateSGPA());
    }

    double calculateSGPA() {
        int totalCredits = 0;
        int totalGradePoints = 0;
        for (int i = 0; i < numSubjects; i++) {
            totalCredits += credits[i];
            int gradePoint = calculateGradePoint(marks[i]);
            totalGradePoints += gradePoint * credits[i];
        }
    }
}

```

```

        return (double) totalGradePoints / totalCredits;

    }

int calculateGradePoint(int marks) {

    if (marks >= 90) return 10;

    else if (marks >= 80) return 9;

    else if (marks >= 70) return 8;

    else if (marks >= 60) return 7;

    else if (marks >= 50) return 6;

    else if (marks >= 40) return 5;

    else return 0;

}

public static void main(String[] args) {

    Scanner sc = new Scanner(System.in);

    System.out.print("Enter number of subjects: ");

    int numSubjects = sc.nextInt();

    Sgpacalc s = new Sgpacalc(numSubjects);

    s.acceptDetails();

    s.displayDetails();

}

```

## OUTPUT:

```
D:\1BM23CS330>java Sgpacalc
Enter number of subjects: 5
Enter USN: 1BM23CS330
Enter Name: Sinchana Hemanth
Enter credits for each subject:
Credits for subject 1: 4
Credits for subject 2: 4
Credits for subject 3: 3
Credits for subject 4: 3
Credits for subject 5: 2
Enter marks for each subject:
Marks for subject 1: 90
Marks for subject 2: 95
Marks for subject 3: 88
Marks for subject 4: 91
Marks for subject 5: 98

Student Details:
USN: 1BM23CS330
Name: Sinchana Hemanth

Subject-wise Credits and Marks:
Subject 1: Credits = 4, Marks = 90
Subject 2: Credits = 4, Marks = 95
Subject 3: Credits = 3, Marks = 88
Subject 4: Credits = 3, Marks = 91
Subject 5: Credits = 2, Marks = 98
SGPA: 9.8125

D:\1BM23CS330>
```

## LABORATORY PROGRAM - 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.

19/10/24

Page No.		
Date		

Program III

Write a java program to create a class Book which contains 4 members : name, author, price, num-pages, Include a constructor to set values for members. Include method to set and get details of objects. Include a `toString()` method that could display complete details of the book. Create n book objects.

```
import java.util.Scanner;
class Book
{
    String name;
    String author;
    double price;
    int numPages;
    Book()
    Book(String name, String author, double price, int numPages)
    {
        this.name = name;
        this.author = author;
        this.price = price;
        this.numPages = numPages;
    }
    void details()
    {
        Scanner sc = new Scanner(System.in);
        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 price:");
        price = sc.nextInt();
        System.out.println("Enter number of pages:");
        pages = sc.nextInt();
    }
}
```

`void display ()`

```
{ System.out.println ("Name of book: " + name);
  System.out.println ("Author of book: " + author);
  System.out.println ("Number of pages: " + numPages);
  System.out.println ("Price: " + price);
```

`}`

`public String toString ()`

`{`

```
  return "Book name: " + name + "\n" + "Author: "
    + author + "\n" + "price: " + price + "\n"
    + "Number of pages: " + numPages;
```

`}`

`class MyBook`

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

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

`System.out.println ("Enter number of books")`  
`int n = sc.nextInt();`

`Book [] books = new Book [n];`

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

`{`

`books[i] = newBook();`

`books[i].details ();`

`}`

`System.out.println ("\nBook Details: ");`

`for (Book book: books)`

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

`}`

3

`3`

14

Page No.	
Date	

Output

Enter the number of books:

2

Enter name of the book:

Silent Patient

Enter name of the author:

Alex Michaelides

Enter price:

499

Enter number of pages

325

Enter name of the book:

~~AGGGTM~~

Enter name of the author:

Holly Jackson

Enter price:

450

Enter number of pages:

400

Book Details :

Book name : Silent Patient

Author : Alex Michaelides

Price : 499

Number of pages : 325

Book name : AGGGTM

Author : Holly Jackson

Price : 450

Number of pages : 400

~~P82~~

24/10/24

## **PROGRAM:**

```
import java.util.Scanner;

class Book

{

    String name;

    String author;

    double price;

    int numPages;

    Book(){ }

    Book(String name,String author,double price,int numPages)

    {

        this.name=name;

        this.author=author;

        this.price=price;

        this.numPages=numPages;

    }

    void details()

    {

        Scanner sc = new Scanner(System.in);

        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 price:");

        price = sc.nextDouble();

    }

}
```

```

        System.out.println("Enter number of pages:");
        numPages = sc.nextInt();
        sc.nextLine();
    }

    void display()
    {
        System.out.println("Name of the book: "+name);
        System.out.println("Author of the book: "+author);
        System.out.println("Price of the book: "+price);
        System.out.println("Number of pages of the book:
"+numPages);
    }

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

class MyBook
{
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter number of books:");
    }
}

```

```

int n = sc.nextInt();

Book[] books = new Book[n];

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

{

    books[i]=new Book();

    books[i].details();

}

System.out.println("\nBook details:");

for(Book book:books)

{

    System.out.println(book); }

}

}

```

## OUTPUT:

```

D:\1BM23CS330>java MyBook
Enter number of books:
2
Enter name of the book:
Silent Patient
Enter name of the author:
Alex Michaelides
Enter price:
499
Enter number of pages:
325
Enter name of the book:
AGGGTM
Enter name of the author:
Holly Jackson
Enter price:
450
Enter number of pages:
400

Book details:
Book name: Silent Patient
Author name: Alex Michaelides
Number of pages: 325
Price of the book: 499.0
Book name: AGGGTM
Author name: Holly Jackson
Number of pages: 400
Price of the book: 450.0

D:\1BM23CS330>

```

## LABORATORY PROGRAM - 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.

21/10/24

Lab program IV

Develop Java program to create abstract class named shape that contains 2 integers and an empty method named printArea(). Provide 3 classes named Rectangle, Triangle and Circle such that each one of the classes extends class Shape. Each one of the classes contain only the method printArea() that prints area of given shape.

```
import java.util.Scanner;
abstract class Shape
{
    float dim1, dim2;
    Shape()
    {
        abstract void printArea();
    }
    class Rectangle extends Shape
    {
        Rectangle()
        {
            void getdim()
            {
                Scanner sc = new Scanner (System.in);
                System.out.println ("Enter length of rectangle:");
                dim1 = sc.nextFloat();
                System.out.println ("Enter breadth of rectangle:");
                dim2 = sc.nextFloat();
            }
            void printArea()
            {
                double area = dim1 * dim2;
                System.out.println ("Area of rectangle: " + area);
            }
        }
    }
}
```

class Triangle extends Shape

1

Triangle () { }

void getd()

2

```
Scanner sc = new Scanner (System.in);
System.out.print("Enter height of triangle:");
dim1 = sc.nextFloat();
System.out.print("Enter base of triangle:");
dim2 = sc.nextFloat();
```

3

void printArea()

4

```
double area = 0.5 * dim1 * dim2;
System.out.print("Area of triangle: "+area);
```

5

class Circle extends Shape

6

Circle () { }

void getd()

7

Scanner sc = new Scanner (System.in);

System.out.print("Enter radius of circle: ");

dim1 = sc.nextFloat();

dim2 = 0.0f;

8

void printArea().

9

double area = Math.PI \* dim1 \* dim1;

System.out.print("Area of circle: "+area);

10

```

class Main {
    public static void main (String [] args) {
        Rectangle rect = new Rectangle ();
        Triangle tri = new Triangle ();
        Circle circ = new Circle ();

        rect.getd();
        tri.getd();
        circ.getd();

        rect.printArea();
        tri.printArea();
        circ.printArea();
    }
}

```

Output :

Enter length of rectangle:

2.5

Enter breadth of rectangle:

2.5

Enter height of triangle:

4.2

Enter base of triangle:

2

Enter radius of circle

5.5

Area of Rectangle : 6.25

Area of Triangle : 4.199999

Area of Circle : 95.03317

Rs. 3  
26/10/24

## **PROGRAM:**

```
import java.util.Scanner;

abstract class Shape

{

    float dim1, dim2;

    abstract void printArea();

}

class Rectangle extends Shape

{

    void getd()

    {

        Scanner sc = new Scanner(System.in);

        System.out.println("Enter length of rectangle:");

        dim1=sc.nextFloat();

        System.out.println("Enter breadth of rectangle:");

        dim2=sc.nextFloat();

    }

    void printArea()

    {

        double area = dim1 * dim2;

        System.out.println("Area of Rectangle: " + area);

    }

}
```

```

class Triangle extends Shape
{
    void getd()
    {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter height of triangle:");
        dim1=sc.nextFloat();
        System.out.println("Enter base of triangle:");
        dim2=sc.nextFloat();
    }

    void printArea()
    {
        double area = 0.5 * dim1 * dim2;
        System.out.println("Area of Triangle: " + area);
    }
}

```

```

class Circle extends Shape
{
    Circle() {}

    void getd()
    {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter radius of circle:");

```

```

        dim1=sc.nextFloat();

        dim2=0.0f;

    }

    void printArea()

    {

        double area = Math.PI * dim1 * dim1;

        System.out.println("Area of Circle: " + area);

    }

}

class Main1

{

    public static void main(String[] args)

    {

        Rectangle rect = new Rectangle();

        Triangle tri = new Triangle();

        Circle circ = new Circle();

        rect.getd();

        tri.getd();

        circ.getd();

        rect.printArea();

        tri.printArea();

        circ.printArea();

    }

}

```

## OUTPUT:

```
D:\1BM23CS330>java Main1
Enter length of rectangle:
2.5
Enter breadth of rectangle:
2.5
Enter height of triangle:
4.2
Enter base of triangle:
2
Enter radius of circle:
5.5
Area of Rectangle: 6.25
Area of Triangle: 4.199999809265137
Area of Circle: 95.03317777109123
D:\1BM23CS330>
```

## **LABORATORY PROGRAM - 5**

Develop a Java program to create a class Bank that maintains two kinds of accounts for its customers, one called a savings account and the other a current account. The savings account provides compound interest and withdrawal facilities but no cheque book facility. The current account provides a 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 deposits from customers 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 a penalty if necessary and update the balance.

Page No.	
Date	

## Lab program II

Develop Java program to create class Bank that maintains 2 kinds of accounts for its customers, one called savings account and other current account. Savings account provides compound interest and withdrawal facilities but no cheque facilities. The current account provides cheque book facilities but no interest. Current account holders should also maintain minimum balance and if balance falls below this level, service charge is imposed.

Create ~~account~~ class Account that stores customer name, account number and type of account. From this derive classes cur-acct and sav-acct to make them more specific to their order to achieve following tasks:

- (a) Accept deposit from customer & update balance
- (b) Display balance
- (c) Compute & deposit interest
- (d) Permit withdrawal & update balance

Check for minimal balance, impose penalty if necessary & update the balance.

```
import java.util.Scanner;
class Account {
    String custName;
    String accNum;
    double deposit;
    double balance;
    double withdrawalAmt;

    void getd() {
        Scanner sc = new Scanner (System.in);
        System.out.println ("Enter customer name");
    }
}
```

```

custName = sc.nextLine();
System.out.println("Enter customer account number");
accNum = sc.nextInt();
System.out.println("Enter deposit amount");
deposit = sc.nextDouble();
balance = deposit;
System.out.println();
}

void putd() {
    System.out.println("Customer name: " + custName);
    System.out.println("Account number: " + accNum);
    System.out.println();
}

class Current extends Account {
    void balanceCheck() {
        if (balance <= 1000) {
            System.out.println("You have less than
minimum balance! 500 is deducted");
            balance -= 500;
        }
    }

    void calcDisplayBalance() {
        System.out.println("Current account details");
        putd();
        System.out.println("Enter amount to be withdrawn");
        Scanner sc = new Scanner(System.in);
        withdrawalAmt = sc.nextDouble();
        balance -= withdrawalAmt;
        balanceCheck();
        System.out.println("Balance (after checking with minimal
balance): " + balance);
        System.out.println();
    }
}

```

```

class SavAcc extends Account {
    void interestCalc() {
        balance = balance + (0.07 * balance);
    }
    void calcDisplayBalance() {
        Scanner sc = new Scanner (System.in);
        System.out.println ("Savings account details");
        putd();
        System.out.println ("Enter amount to be withdrawn");
        withdrawalAmt = sc.nextDouble();
        balance -= withdrawalAmt;
        System.out.println ("Balance before addition of
                           interest: " + balance);
        interestCalc();
        System.out.println ("Balance after addition of
                           interest: " + balance);
        System.out.println ();
    }
}

class Bank {
    public static void main (String args[]) {
        Scanner sc = new Scanner (System.in);
        String accType;
        System.out.println ("Enter account type (savings
                           account / current account)");
        accType = sc.nextLine();
        if (accType.equals ("savings account")) {
            SavAcc sacc = new SavAcc();
            sacc.getd();
        }
    }
}

```

sacc. calcDisplayBalance();

3

else if (acctType.equals("Current account")) {

CurrAcc cacc = new CurrAcc();

cacc.getBal();

cacc. calcDisplayBalance();

9

else {

System.out.println("Enter a valid account type");

9

3

### Output

Enter type of account (Savings account / Current account)

Savings account

Enter customer name

Srichana

Enter customer account number

123AB43C

Enter deposit amount

25000

Savings account details:

Customer name: Srichana

Account number: 123AB43C

Enter amount to be withdrawn

10000

Balance before addition of interest: 15000.0

Balance after addition of interest: 16050.0

## **PROGRAM:**

```
import java.util.Scanner;

class Account {

    String custName;
    String accNum;
    double deposit;
    double balance;
    double withdrawalAmt;

    void getd() {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the customer name");
        custName = sc.nextLine();
        System.out.println("Enter the customer account number");
        accNum = sc.nextLine();
        System.out.println("Enter the deposit amount");
        deposit = sc.nextDouble();
        balance=deposit;
        System.out.println();
    }

    void putd() {
        System.out.println("Customer name: "+custName);
        System.out.println("Account number: "+accNum);
        System.out.println();
    }
}
```

```

        }

    }

class CurAcct extends Account {

    void balanceCheck() {

        if (balance<=1000) {

            System.out.println("You have less than
minimum balance! 500rs deducted");

            balance-=500;

        }

    }

    void calcDisplayBalance() {

        System.out.println("Current account details");

        putd();

        System.out.println("Enter amount to be withdrawn");

        Scanner sc = new Scanner(System.in);

        withdrawalAmt = sc.nextDouble();

        balance-=withdrawalAmt;

        balanceCheck();

        System.out.println("Balance (after checking with
minimum balance): "+balance);

        System.out.println();

    }

}

class SavAcct extends Account {

```

```

void interestCalc() {
    balance=balance+(0.07*balance);
}

void calcDisplayBalance() {
    Scanner sc = new Scanner(System.in);
    System.out.println("Savings account details");
    putd();
    System.out.println("Enter amount to be withdrawn");
    withdrawalAmt = sc.nextDouble();
    balance-=withdrawalAmt;
    System.out.println("Balance before addition of compound interest:
"+balance);
    interestCalc();
    System.out.println("Balance after compound interest addition:
"+balance);
    System.out.println();
}

```

```

class Bank {
    public static void main(String args[]) {
        Scanner sc = new Scanner(System.in);
        String accType;
        System.out.println("Enter the type of account (Savings
account or Current account)");
        accType = sc.nextLine();
    }
}

```

```
if (accType.equals("Savings account")) {  
    SavAcct sacc = new SavAcct();  
    sacc.getd();  
    sacc.calcDisplayBalance();  
}  
  
else if (accType.equals("Current account")) {  
    CurAcct cacc = new CurAcct();  
    cacc.getd();  
    cacc.calcDisplayBalance();  
}  
  
else {  
    System.out.println("Enter a valid account type");  
}  
}
```

## OUTPUT:

```
D:\1BM23CS330>java Bank
Enter the type of account (Savings account or Current account)
Savings account
Enter the customer name
Sinchana Hemanth
Enter the customer account number
123AB43C
Enter the deposit amount
25000

Savings account details
Customer name: Sinchana Hemanth
Account number: 123AB43C

Enter amount to be withdrawn
10000
Balance before addition of compound interest: 15000.0
Balance after compound interest addition: 16050.0

D:\1BM23CS330>java Bank
Enter the type of account (Savings account or Current account)
Current account
Enter the customer name
Sinchana Hemanth
Enter the customer account number
567NMD05
Enter the deposit amount
15000

Current account details
Customer name: Sinchana Hemanth
Account number: 567NMD05

Enter amount to be withdrawn
14000
You have less than minimum balance! 500rs deducted
Balance (after checking with minimum balance): 500.0
```

## LABORATORY PROGRAM - 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.

Page No. \_\_\_\_\_  
Date \_\_\_\_\_

Lab program VI

Create a package CIE which has 2 classes - Student & Internals. Class Student has members like usn & name. Class Internals has an array that stores the internal marks scored in 5 courses of current Semester of student. Create another package SEE which has class External which is derived class of Student. This class has an array that stores SEE marks scored in 5 courses of current semester of student. Import the 2 packages in a file & declare final marks of n students in all 5 courses.

```
package CIE;
import java.util.Scanner;
public class Student {
    String name;
    String usn;
    int sem;
    public void getd() {
        Scanner sc = new Scanner (System.in);
        System.out.println ("Enter student USN ");
        usn = sc.nextLine();
        System.out.println ("Enter student name ");
        name = sc.nextLine();
        System.out.println ("Enter the semester ");
        sem = sc.nextInt();
    }
    public void display () {
        System.out.println ();
        System.out.println ("Student USN : " + usn);
        System.out.println ("Student name : " + name);
        System.out.println ("Semester : " + sem);
        System.out.println ();
    }
}
```

```

package CIE;
import java.util.Scanner;
public class Internals {
    public int markscie[] = new int[5];
    public void getMarks() {
        for (int i=0; i<5; i++) {
            Scanner sc = new Scanner(System.in);
            System.out.println("Enter CIE marks in
                subject " + (i+1));
            markscie[i] = sc.nextInt();
        }
    }
    public int returnMarkscie (int i) {
        return markscie[i];
    }
}

```

```

package SEE;
import CIE.Student;
import CIE.Internals;
import java.util.Scanner;
public class Externals extends Student {
    int markssee[] = new int[5]
    public void getMarks() {
        for (int i=0; i<5; i++) {
            Scanner sc = new Scanner(System.in);
            System.out.println("Enter SEE marks in
                subject " + (i+1));
            markssee[i] = sc.nextInt();
        }
    }
    public void calculateMarks (Internals i) {
        for (int l=0; l<5; l++) {

```

```
System.out.println ("Subject " + (i+1) + ":" +  
    (l1. returnMarksie (l) + (marksse [l] / 2)));
```

}

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

g

j

```
import CIE.Student;
```

```
import CIE.Internals;
```

```
import SEE.Externals;
```

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

```
public class Main {
```

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

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

```
System.out.println ("Enter number of students")
```

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

```
Internals [] li = new Internals [n];
```

```
Externals [] ei = new Externals [n];
```

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

```
System.out.println ("Student " + (i+1) +
```

```
" details:") ;
```

```
ei[i] = new Externals();
```

```
li[i] = new Internals();
```

```
ei[i].getdetails();
```

```
li[i].getmarks();
```

```
ei[i].getmarks();
```

g

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

```
ei[i].display();
```

```
ei[i].calcTotalMarks (li[i]);
```

g

j

	<u>Output:</u>
	Enter the number of students
	2
	Student 1 details:
	Enter student USN
	1BN123CS330
	Enter student name
	Sinchana Hemanth
	Enter semester
	3
	Enter CIE marks in subject 1
	45
	Enter CIE marks in subject 2
	49
	Enter CIE marks in subject 3
	48
	Enter CIE marks in subject 4
	40
	Enter CIE marks in subject 5
	50
	Enter SEE marks in subject 1
	95
	Enter SEE marks in subject 2
	98
	Enter SEE marks in subject 3
	90
	Enter SEE marks in subject 4
	100
	Enter SEE marks in subject 5
	100
	Student 2 details:

Page No.	
Date:	

Enter student USN

18M23CS317

Enter student name

Sriya Raj

Enter semester

3

Enter CIE marks in subject 1

48

Enter CIE marks in subject 2

40

Enter CIE marks in subject 3

50

Enter CIE marks in subject 4

45

Enter CIE marks in subject 5

42

Enter SEE marks in subject 1

100

Enter SEE marks in subject 2

90

Enter SEE marks in subject 3

96

Enter SEE marks in subject 4

92

Enter SEE marks in subject 5

90

Student USN: 18M23CS330

Student name: Srihanna Hemanth

Semester: 3

Subject 1: 92

Subject 2: 98

Subject 3: 93

Subject 4: 90

Subject 5: 100

Student USN: IBM123CS317

Student name: Shourya Raj

Semester: 3

Subject 1: 98

Subject 2: 85

Subject 3: 98

Subject 4: 91

Subject 5: 87

Raj

1911124

## **PROGRAM:**

(FOLDER 1: CIE)

(FILE 1: INTERNALS)

```
package CIE;  
import java.util.Scanner;  
  
public class Internals {  
  
    public int marksCie[] = new int[5];  
  
    public void getMarks() {  
  
        for(int i=0;i<5;i++) {  
  
            Scanner sc = new Scanner(System.in);  
  
            System.out.println("Enter CIE marks in subject  
"+(i+1));  
  
            marksCie[i]=sc.nextInt();  
  
        }  
  
    }  
  
    public int returnCieMarks(int i) {  
  
        return marksCie[i];  
  
    }  
}
```

(FILE 2: STUDENT)

```
package CIE;  
import java.util.Scanner;  
  
public class Student {  
  
    String usn;  
  
    String name;
```

```

int sem;

public void getd() {

    Scanner sc = new Scanner(System.in);

    System.out.println("Enter student USN");

    usn = sc.nextLine();

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

    name = sc.nextLine();

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

    sem = sc.nextInt();

}

public void display() {

    System.out.println();

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

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

    System.out.println("Semester: "+sem);

    System.out.println();

}

}

```

(FOLDER 2: SEE)

(FILE 1: EXTERNALS)

```

package SEE;

import CIE.Student;

import CIE.Internals;

import java.util.Scanner;

```

```

public class Externals extends Student {
    int marksSee[] = new int[5];
    public void getMarks() {
        for(int i=0;i<5;i++) {
            Scanner sc = new Scanner(System.in);
            System.out.println("Enter SEE marks in subject "+(i+1));
            marksSee[i]=sc.nextInt();
        }
    }
    public void calcTotalMarks(Internals i1) {
        for(int i=0;i<5;i++) {
            System.out.println("Subject "+(i+1)+": "+(i1.returnCieMarks(i)+(marksSee[i]/2)));
        }
        System.out.println();
    }
}

```

### (MAIN CLASS)

```

import CIE.Student;
import CIE.Internals;
import SEE.Externals;
import java.util.Scanner;
public class Main {

```

```

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

    System.out.println("Enter the number of students
whose details you want to enter");

    int n = sc.nextInt();

    Internals[] i1 = new Internals[n];
    Externals[] e1 = new Externals[n];

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

        System.out.println("Student "+(i+1)+" details:");
        e1[i] = new Externals();
        i1[i] = new Internals();
        e1[i].getd();
        i1[i].getMarks();
        e1[i].getMarks();

    }

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

        e1[i].display();
        e1[i].calcTotalMarks(i1[i]);

    }

}

```

## OUTPUT:

```
D:\package>java Main
Enter the number of students whose details you want to enter
2
Student 1 details:
Enter student USN
1BM23CS330
Enter student name
Sinchana Hemanth
Enter semester
3
Enter CIE marks in subject 1
45
Enter CIE marks in subject 2
49
Enter CIE marks in subject 3
48
Enter CIE marks in subject 4
40
Enter CIE marks in subject 5
50
Enter SEE marks in subject 1
95
Enter SEE marks in subject 2
98
Enter SEE marks in subject 3
90
Enter SEE marks in subject 4
100
Enter SEE marks in subject 5
100
Student 2 details:
Enter student USN
1BM23CS317
Enter student name
Shreya Raj
Enter semester
3
Enter CIE marks in subject 1
48
Enter CIE marks in subject 2
40
Enter CIE marks in subject 3
50
Enter CIE marks in subject 4
45
Enter CIE marks in subject 5
42
Enter SEE marks in subject 1
100
Enter SEE marks in subject 2
90
Enter SEE marks in subject 3
96
Enter SEE marks in subject 4
92
Enter SEE marks in subject 5
90
```

## LABORATORY PROGRAM - 7

Write a program that demonstrates handling of exceptions in inheritance tree. 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.

Page No. \_\_\_\_\_  
Date \_\_\_\_\_

Lab program VII

Write a java program that demonstrates handling of exceptions in inheritance tree. Create a base class called "Father" and derived class called "son" which extends base class. In father's class , implement constructor which takes age & throws exception wrongage() when input age<0. In son's class , implement constructor that uses both father's & son's age and throws exception if son's age is  $\geq$  father's age

```
import java.util.Scanner;
class WrongAge extends Exception {
    String message;
    WrongAge (String message) {
        this, message = message;
    }
    public String toString() {
        return "Error: "+ message;
    }
}
class Father {
    int age;
    Father (int x) throws WrongAge {
        if (x<0) {
            throw new WrongAge("Father's age cannot be negative");
        }
        age = x;
    }
}
```

```

class Son extends Father {
    int sage;
    Son (int x, int y) throws WrongAge {
        super(x);
        if (y < 0) {
            throw new WrongAge ("Son's age cannot be
negative");
        }
        if (y >= x) {
            throw new WrongAge ("Son's age cannot be
greater than Father's age");
        }
        sage = y;
    }
}

```

```

class Except {
    public static void main (String args[]) {
        Scanner sc = new Scanner (System.in);
        try {
            int x, y;
            System.out.println ("Enter Father's age");
            x = sc.nextInt();
            System.out.println ("Enter son's age");
            y = sc.nextInt();
            Son son = new Son (x, y);
            System.out.println ("Father's age: " + son.father);
            System.out.println ("Son's age: " + son.sage);
        }
        catch (WrongAge wa) {
            System.out.println (wa);
        }
    }
}

```

Page No.	
Date	

Output I

Enter Father's age

~~40~~

Enter son's age

5

Father's age : ~~40~~

Son's age : 5

Output II

Enter Father's age

-2

Enter son's age

12

Error: Father's age cannot

Output III

Enter Father's age

25

Enter son's age

-8

Son's age cannot be  
negative

Output IV

Enter Father's age

15

Enter son's age

12

Error: Son's age cannot be  
greater than Father's age

~~12~~

~~21/11/24~~

2;

## **PROGRAM:**

```
import java.util.Scanner;

class WrongAge extends Exception {

    String message;

    WrongAge (String message) {
        this.message = message;
    }

    public String toString() {
        return "Error: " + message;
    }
}

class Father {

    int fage;

    Father(int x) throws WrongAge {
        if(x<0) {
            throw new WrongAge("Father's age cannot be
negative");
        }
        fage = x;
    }
}

class Son extends Father {

    int sage;
```

```

Son(int x,int y) throws WrongAge {
    super(x);
    if(y<0) {
        throw new WrongAge("Son's age cannot be
negative");
    }
    if(y>=x) {
        throw new WrongAge("Son's age cannot be
greater than Father's age");
    }
    sage = y;
}

```

```

class Excep {
    public static void main(String args[]) {
        Scanner sc = new Scanner(System.in);
        try {
            int x,y;
            System.out.println("Enter father's age");
            x=sc.nextInt();
            System.out.println("Enter son's age");
            y=sc.nextInt();
            Son son = new Son(x,y);
            System.out.println("Father's age: "+son.fage);
            System.out.println("Son's age: "+son.sage);
        }
    }
}

```

```

        }

    catch (WrongAge wa) {
        System.out.println(wa);

    }

}

```

## OUTPUT:

```

D:\1BM23CS330>java Excep
Enter father's age
40
Enter son's age
5
Father's age: 40
Son's age: 5

D:\1BM23CS330>java Excep
Enter father's age
-2
Enter son's age
12
Error: Father's age cannot be negative

D:\1BM23CS330>java Excep
Enter father's age
25
Enter son's age
-8
Error: Son's age cannot be negative

D:\1BM23CS330>java Excep
Enter father's age
45
Enter son's age
12
Father's age: 45
Son's age: 12

```

## LABORATORY PROGRAM - 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.

Lab program VIII

Write a Java program which creates two threads, one thread displaying "BMS college of Engineering" once every 10 seconds and another displaying "CSE" every 2 seconds.

```
class A extends Thread {  
    private boolean running = true;  
    public void terminate() {  
        running = false;  
    }  
    public void run() {  
        while (running) {  
            System.out.println("BMS college of Engineering");  
            try {  
                Thread.sleep(10000);  
            } catch (InterruptedException ie) {  
                System.out.println("Sleeping thread woken up");  
            }  
        }  
    }  
}  
  
class B extends Thread {  
    private boolean running = true;  
    public void terminate() {  
        running = false;  
    }  
    public void run() {  
        while (running) {  
            System.out.println("CSE");  
        }  
    }  
}
```

try {

Thread.sleep(2000);

}

catch (InterruptedException ie) {

System.out.println ("Sleeping thread woken  
up");

}

}

}

~~public class~~ ThreadExample {

public static void main (String args[]) {

};

A threadA = new A();

B threadB = new B();

threadA.start();

threadB.start();

try {

woken up);

Thread.sleep(30000);

}

catch (InterruptedException ie) {

System.out.println ("Main thread interrupt  
ed");

}

threadA.terminate();

threadB.terminate();

System.out.println ("Program terminated");

3

Page No.	
Date	

Output

BMS College of Engineering

CSE

CSE

CSE

CSE

CSE

BMS College of Engineering

CSE

BMS College of Engineering

Program terminated

11/11/2021  
29/11/2021

## **PROGRAM:**

```
class A extends Thread {  
    public void run() {  
        while(true) {  
            System.out.println("BMS College of Engineering");  
            try {  
                Thread.sleep(10000);  
            }  
            catch (InterruptedException ie) {  
                System.out.println("Thread has been woken up");  
            }  
        }  
    }  
}
```

```
class B extends Thread {  
    public void run() {  
        while(true) {  
            System.out.println("CSE");  
            try {  
                Thread.sleep(2000);  
            }  
            catch (InterruptedException ie) {  
                System.out.println("Thread has been woken up");  
            }  
        }  
    }  
}
```

```

        }

    }

}

class Main2 {

    public static void main(String args[]) {

        A objA = new A();

        B objB = new B();

        objA.start();

        objB.start();

    }

}

```

### **OUTPUT:**

```

D:\1BM23CS330>java Main2
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
^C
D:\1BM23CS330>

```

## LABORATORY PROGRAM - 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.

Page No. \_\_\_\_\_  
Date: \_\_\_\_\_

Lab program IX

Write a program that creates user interface to perform integer division. User enters 2 numbers in text fields, NUM1 & NUM2. The division of NUM1 & NUM2 is displayed in result field when Divide button is clicked. If NUM1 & NUM2 were not integer the program would throw NumberFormatException. If NUM2 were 0 program would throw ArithmeticException. Display the exception in message dialog box.

```
import java.awt.*;
import java.awt.event.*;
public class Divisionmain extends Frame implements
ActionListener
{
    TextField num1, num2;
    Button dresult;
    Label aresult;
    String str = " ";
    double resultNum;
    int flag = 0;
    public Divisionmain()
    {
        setLayout(new FlowLayout());
        dresult = new Button("RESULT");
        Label number1 = new Label("Number 1: ", Label.RIGHT);
        Label number2 = new Label("Number 2: ", Label.RIGHT);
        num1 = new TextField(5);
        num2 = new TextField(5);
        aresult = new Label("Result: ", Label.RIGHT);
        add(number1);
        add(num1);
        add(number2);
        add(num2);
        add(dresult);
        add(aresult);
        dresult.addActionListener(this);
    }
    public void actionPerformed(ActionEvent e)
    {
        if(flag == 0)
        {
            try
            {
                resultNum = Double.parseDouble(num1.getText());
                resultNum = resultNum / Double.parseDouble(num2.getText());
                aresult.setText(str + resultNum);
            }
            catch(NumberFormatException nfe)
            {
                JOptionPane.showMessageDialog(null, "Please enter integer values");
            }
            catch(ArithmeticException ae)
            {
                JOptionPane.showMessageDialog(null, "Division by zero is not allowed");
            }
        }
        else
        {
            JOptionPane.showMessageDialog(null, "Please enter integer values");
        }
    }
}
```

```

    add(num1);
    add(number2);
    add(num2);
    add(dResult);
    add(outResult);
    num1.addActionListener(this);
    num2.addActionListener(this);
    dResult.addActionListener(this);
    addWindowListener(new WindowAdapter() {
        public void windowClosing(WindowEvent we) {
            System.exit(0);
        }
    });
    public void actionPerformed(ActionEvent ae) {
        int n1, n2;
        try {
            if (ae.getSource() == dResult) {
                n1 = Integer.parseInt(num1.getText());
                n2 = Integer.parseInt(num2.getText());
                /* if (n2 == 0)
                   throw new ArithmeticException(); */
                out = n1 + " " + n2 + "=";
                resultNum = n1 / n2;
                out += String.valueOf(resultNum);
                repaint();
            }
        }
        catch (NumberFormatException e1) {
            flag = 1;
            out = "Number Format Exception! " + e1;
            repaint();
        }
    }
}

```

```
catch (ArithmeticException e2) {
```

```
    flag = 1;
```

```
    out = " Divide by 0 Exception " + e2;
```

```
    repaint();
```

g

```
public void paint (Graphics g) {
```

```
    if (flag == 0) {
```

```
        g.drawString (out, outResult.getX() +
```

```
                     outResult.getWidth(), outResult.getY()
```

```
() + outResult.getHeight() - 8);
```

~~else~~

g

```
else {
```

```
    g.drawString (out, 100, 200);
```

```
    flag = 0;
```

g

Output

Division of Integers

Number 1: 10 Number 2: 5 Result:  $10/5 = 2.0$

## **PROGRAM:**

```
import java.awt.*;
import java.awt.event.*;
public class DivisionMain1 extends Frame implements ActionListener {
    TextField num1,num2;
    Button dResult;
    Label outResult; String out="";
    double resultNum; int flag=0;
    public DivisionMain1() {
        setLayout(new FlowLayout());
        dResult = new Button("RESULT");
        Label number1 = new Label("Number 1:",Label.RIGHT);
        Label number2 = new Label("Number 2:",Label.RIGHT);
        num1=new TextField(5);
        num2=new TextField(5);
        outResult = new Label("Result:",Label.RIGHT);
        add(number1);
        add(num1);
        add(number2);
        add(num2);
        add(dResult);
        add(outResult);
        num1.addActionListener(this);
        num2.addActionListener(this);
        dResult.addActionListener(this);
        addWindowListener(new WindowAdapter() {
            public void windowClosing(WindowEvent we) {
```

```

        System.exit(0);
    }
});

}

public void actionPerformed(ActionEvent ae) {
    int n1,n2;
    try {
        if (ae.getSource() == dResult) {
            n1=Integer.parseInt(num1.getText());
            n2=Integer.parseInt(num2.getText());
            out=n1+" "+n2+" ";
            resultNum=n1/n2;
            out+=String.valueOf(resultNum);
            repaint();
        }
    }
    catch(NumberFormatException e1) {
        flag=1;
        out="Number Format Exception! "+e1;
        repaint();
    }
    catch(ArithmaticException e2) {
        flag=1;
        out="Divide by 0 Exception! "+e2;
        repaint();
    }
}

public void paint(Graphics g) {

```

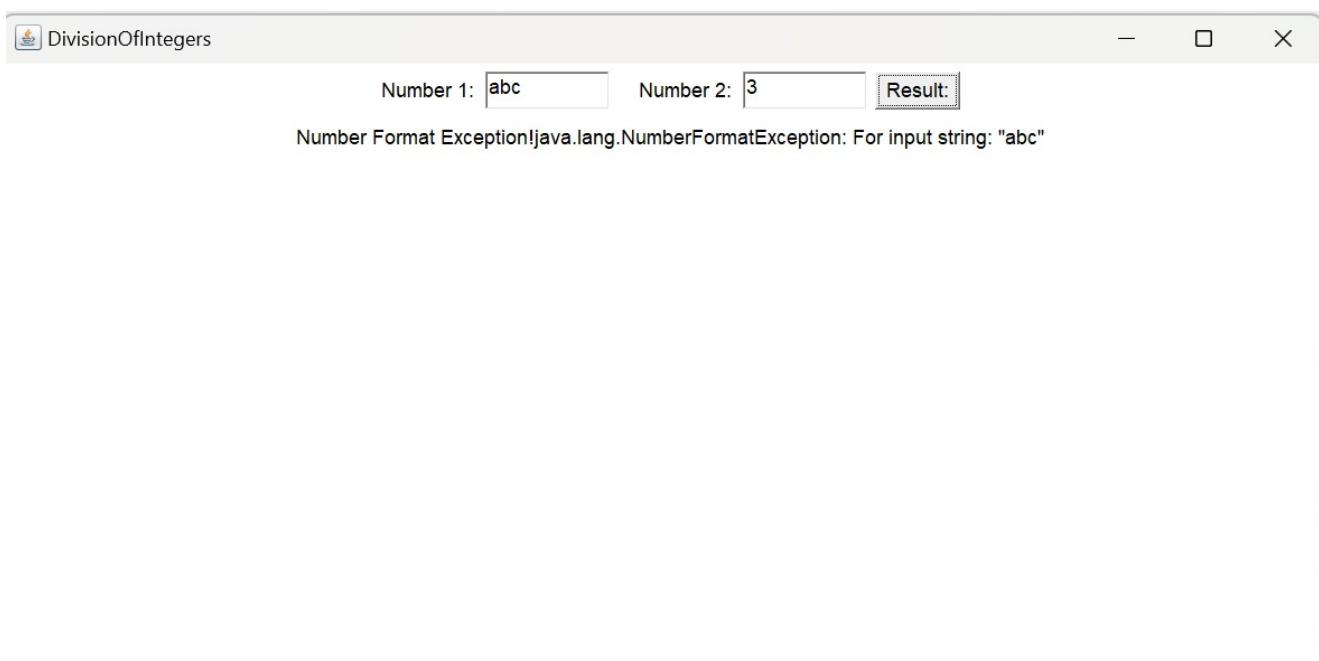
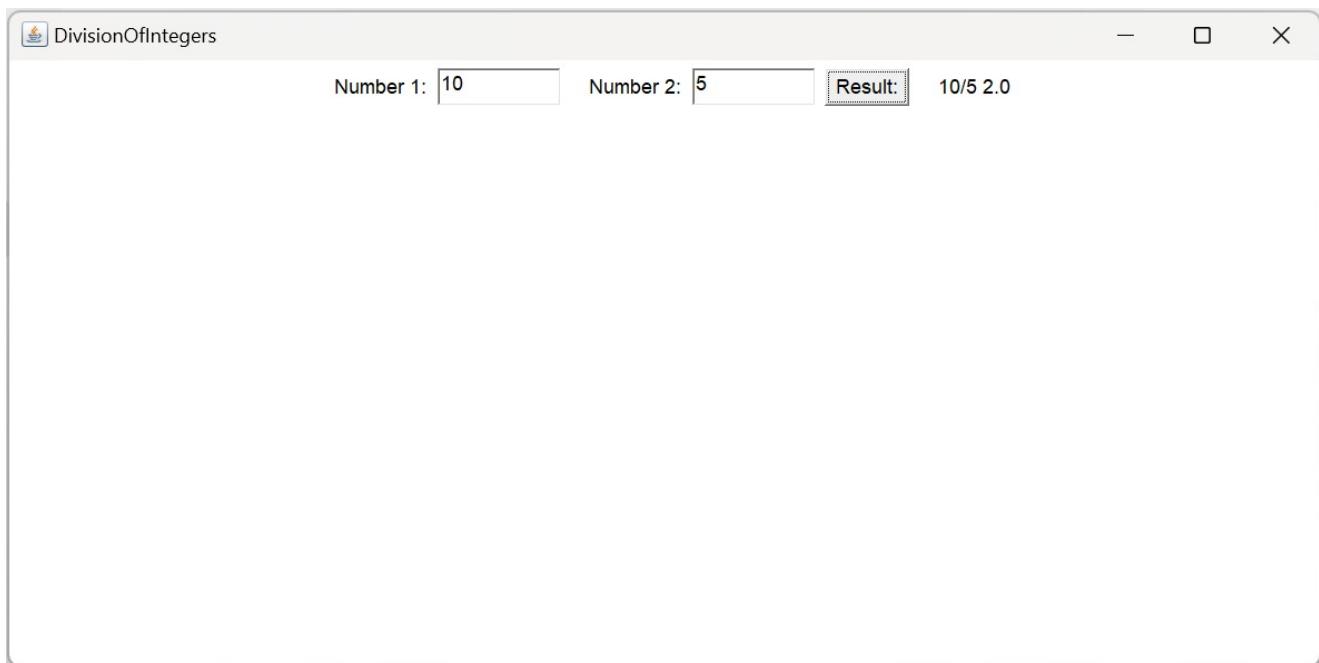
```
if(flag==0)
g.drawString(out,outResult.getX()+outResult.getWidth(),outResult.getY()+outResult.getHeight()-8);

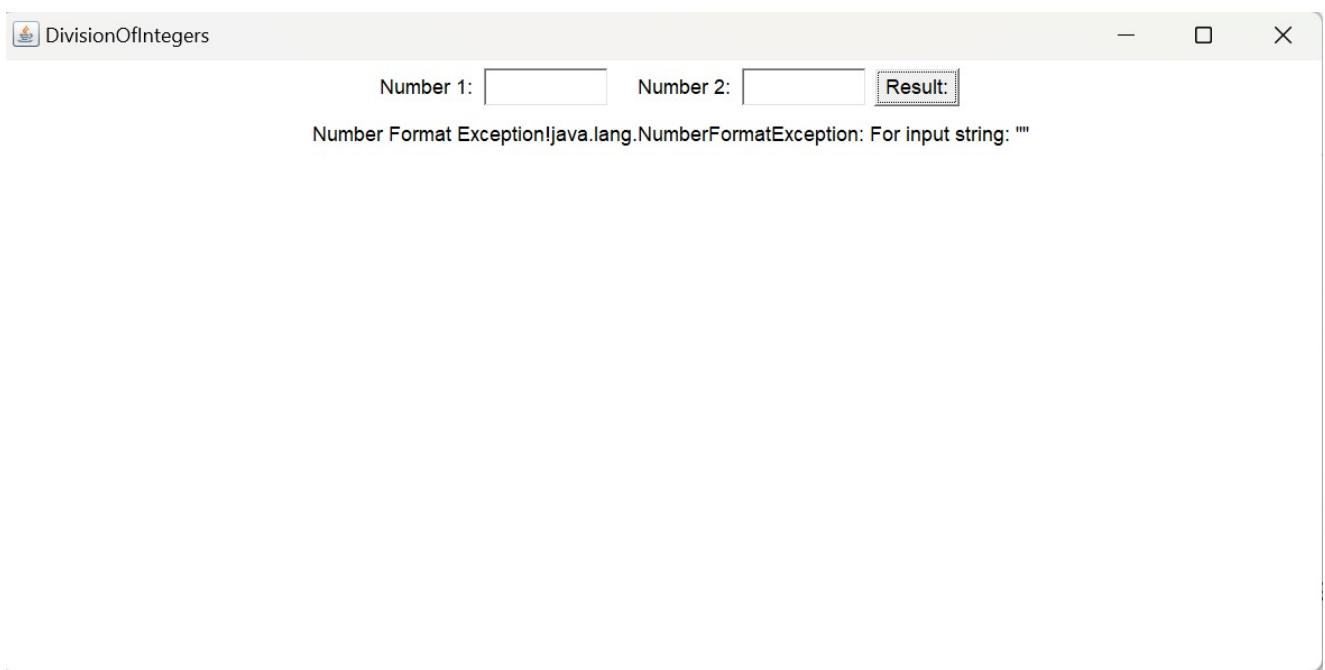
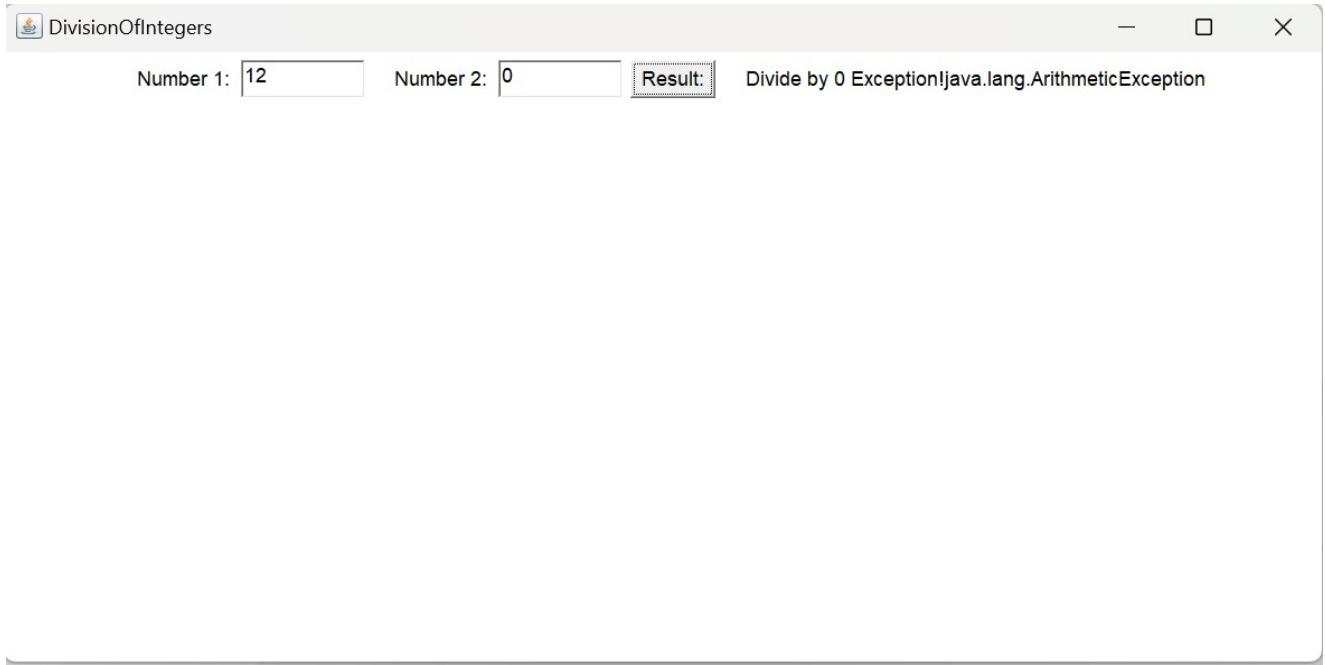
else g.drawString(out,100,200);

flag=0;

}
```

## OUTPUT:





## LABORATORY PROGRAM - 10

Write Demonstrate Interprocess communication and deadlock

(i) Interprocess communication:

Page No. \_\_\_\_\_  
Date \_\_\_\_\_

```
this.n = n;  
valueset = true;  
System.out.println("Put: "+n);  
System.out.println("In Indimate Consumer(n);  
notify();  
}  
}  
  
class Producer implements Runnable {  
    Queue 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 {  
    Queue q;  
    consumer(Q q) {  
        this.q = q;  
        new Thread(this, "consumer").start();  
    }  
    public void run() {  
        int l=0;  
        while(l<15) {  
            int n = q.get();  
        }  
    }  
}
```

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

i++;

}

g

3

class PCFinal {

public static void main (String [] args) {

Q q = new Q();

new Producer (q);

new Consumer (q);

System.out.println ("Press Control-C to stop.");

g

g

### Output

Press control-C to stop

Put: 0

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 Producer

Put: 14

consumed 14

Intimate consumer

Got: 14



## **PROGRAM:**

```
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");  
            }  
            valueSet = true;  
            System.out.println("Put: " + n);  
            notify();  
        }  
    }  
}
```

```

        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.");  
    }  
}
```

## OUTPUT:

```
D:\1BM23CS330>java PCFixed
Press Control-C to stop.
Put: 0

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: 5
Intimate Producer
consumed:5
Put: 6
Intimate Consumer

Producer waiting
Got: 6
Intimate Producer
consumed:6
Put: 7
Intimate Consumer

Producer waiting
Got: 7
Intimate Producer
consumed:7
Put: 8
```

```
Intimate Consumer

Producer waiting
Got: 8
Intimate Producer
consumed:8
Put: 9
Intimate Consumer

Producer waiting
Got: 9
Intimate Producer
consumed:9
Put: 10
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

D:\1BM23CS330>
```

(i) Demonstration of deadlock:

Page No.	
Date	

(ii) Demonstration of 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();  
    }  
}
```

synchronized void last() {

```
    System.out.println("Inside A.last");  
}
```

}

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();  
}
```

synchronized void last() {

```
    System.out.println("Inside A.last");  
}
```

3

Print No.	
Date	

class Deadlock implements Runnable {

A a = new A();

B b = new B();

Deadlock() {

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

Thread t = new Thread (this, "RacingThread");

t.start();

a.foo(b);

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

}

public void run() {

b.bar(a);

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

}

public static void main (String [] args) {

new Deadlock();

}

public static void main (String [] args) {

DivisionMain dm = new DivisionMain();

dm.setSize(new Dimension(800,400));

dm.setTitle ("Division of Integers");

dm.setVisible(true);

}

}

### Output

RacingThread entered B.bar

MainThread entered A.foo

RacingThread trying to call A.last()

MainThread trying to call B.last()

## **PROGRAM:**

```
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();
        synchronized 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();
            synchronized 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);
            }
        }
    }
}
```

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

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

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

}
```

## OUTPUT:

```
D:\1BM23CS330>javac Deadlock.java

D:\1BM23CS330>java Deadlock
RacingThread entered B.bar
MainThread entered A.foo
RacingThread trying to call A.last()
MainThread trying to call B.last()
```