

**VISVESVARAYA TECHNOLOGICAL  
UNIVERSITY**

“JnanaSangama”, Belgaum -590014, Karnataka.



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

*Submitted by*

*Ameena Yasmeen (1BM23CS027)*

*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 **Ameena Yasmeen(1BM23CS027)**, who is 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.

Lab faculty Incharge Name Assistant Professor Department of CSE, BMSCE	Basavaraj Jakkali. Professor & HOD Department of CSE, BMSCE
--	---

## Index

<u>Sl. No.</u>	<u>Date</u>	<u>Experiment Title</u>	<u>Page No.</u>
1	9/10/24	Implement Quadratic Equation	4-6
2	16/10/24	Implement SGPA Calculator	6-11
3	23/10/24	Create Objects For Books	11-16
4	30/10/24	Implement Abstract Class	17-21
5	6/11/24	Bank Account Management	21-27
6	13/11/24	Implement Packages	28-39
7	20/11/24	Implement Exception Handling	40-50
8	27/11/24	Multithreading, Creating Threads In Java	51-65
9	27/11/24	Interface To Perform Integer Division	66-72
10	27/11/24	Implement Deadlock	73-86

<https://github.com/Ameena1BM23CS27/java-lab/tree/main/Java%20lab%201>

## Program 1

### Implement Quadratic Equation

Date \_\_\_\_\_  
Page \_\_\_\_\_

```
import java.util.Scanner;
import java.lang.Math;

class quad{
    public static void main (String [] args) {
        Scanner scanner = new Scanner (System.in);
        System.out.println ("Enter coefficient of a:");
        double a = scanner.nextDouble();

        System.out.println ("Enter coefficient of b:");
        double b = scanner.nextDouble();

        System.out.println ("Enter coefficient of c:");
        double c = scanner.nextDouble();

        double d = b * b - 4 * a * c;
        if (d > 0) {
            double r1 = (-b + Math.sqrt(d)) / (2 * a);
            double r2 = (-b - Math.sqrt(d)) / (2 * a);

            System.out.println ("The roots are real and
                different");
            System.out.println ("Root 1: " + r1);
            System.out.println ("Root 2: " + r2);
        }
        else if (d == 0) {
            double r = -b / (2 * a);
            System.out.println ("There is one real solution:
                x = " + r);
        }
    }
}
```

```
else {
    double realpart = -b / (2 * a);
    double imaginarypart = Math.sqrt(-d) / (2 * a);
```

```
System.out.println("The roots are complex");
```

```
System.out.println("Root 1: " + realpart + " + "
    + imaginarypart + "i");
```

```
System.out.println("Root 2: " + realpart + " - "
    + imaginarypart + "i");
```

}

```
System.out.println("Ameena Yasmeen \n IBN23CS027");
```

}

Output

Enter coefficient of a:

3

Enter coefficient of b:

6

Enter coefficient of c:

6

The roots are complex

Root 1: 1.0 + 0.0i

Root 2: -1.0 - 0.0i

Ameena Yasmeen IBN23CS027

Ras

6/10/24

## Source Code:

```
import java.util.Scanner;

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

        System.out.println("");
        System.out.print("Enter coefficient a: ");
        double a = scanner.nextDouble();
        System.out.print("Enter coefficient b: ");
        double b = scanner.nextDouble();
        System.out.print("Enter coefficient c: ");
        double c = scanner.nextDouble();

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

        if (a == 0) {
            System.out.println("This is not a quadratic equation (a cannot
be zero).");
        } else {
```

```
if (discriminant > 0) {  
    double root1 = (-b + Math.sqrt(discriminant)) / (2 * a);  
    double root2 = (-b - Math.sqrt(discriminant)) / (2 * a);  
    System.out.println("The roots are real and distinct:");  
    System.out.println("Root 1: " + root1);  
    System.out.println("Root 2: " + root2);  
}  
else if (discriminant == 0) {  
    double root = -b / (2 * a);  
    System.out.println("The roots are real and equal:");  
    System.out.println("Root: " + root);  
}  
else {  
    double realPart = -b / (2 * a);  
    double imaginaryPart = Math.sqrt(-discriminant) / (2 * a);  
    System.out.println("The roots are complex:");  
    System.out.println("Root 1: " + realPart + " + " +  
        imaginaryPart + "i");  
    System.out.println("Root 2: " + realPart + " - " +  
        imaginaryPart + "i");  
}  
}  
scanner.close();  
}
```

## **Output:**

```
D:\>javac QuadraticEquation.java
D:\>java QuadraticEquation
Ameena Yasmeen
1BM23CS027
Enter coefficient a: 4
Enter coefficient b: 5
Enter coefficient c: 6
The roots are complex:
Root 1: -0.625 + 1.0532687216470449i
Root 2: -0.625 - 1.0532687216470449i
```

<https://github.com/Ameena1BM23CS27/java-lab/commit/6ed0a38366980ad8e1173c9918e6a245ab47057>  
2

## Program 2

### Implement SGPA Calculator

Lab 2:-  
Lab 2  
Date \_\_\_\_\_  
Page \_\_\_\_\_  
16/10

```
import java.util.Scanner;

class Student {
    String usn;
    String name;
    int [] credits;
    int [] marks;
}

public Student(int numSubjects) {
    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 and marks for each subjects:");
    for (int i=0; i< credits.length; i++) {
        System.out.print ("Credits for subjects " + (i+1));
        credits[i] = sc.nextInt();
        System.out.print ("Marks for Subject " + (i+1) + ":");
        marks[i] = sc.nextInt();
    }
}
```

```
void displayDetails() {
```

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

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

```
    for (int i = 0; i < credits.length; i++) {
```

```
        System.out.println("Subject " + (i + 1) + ". Credits: " + credits[i] + ", Marks: " + marks[i]);
```

```
}
```

```
System.out.print("SGPA: %.2f\n", calculateSGPA());
```

```
}
```

```
double calculateSGPA() {
```

```
    double total credits = 0;
```

```
    double total points = 0;
```

```
    for (int i = 0; i < credit.length; i++) {
```

```
        double gradePoint = calculateGradePoint(marks[i]);
```

```
        total points += gradePoint * credit[i];
```

```
        total credit += credits[i];
```

```
} return total credits > 0 ? total points / total credits : 0;
```

```
double calculateGradePoint(int mark) {
```

```
    if (mark >= 90) return 10;
```

```
    else if (mark >= 80) return 9;
```

```
    else if (mark >= 70) return 8;
```

```
    else if (mark >= 60) return 7;
```

```
    else if (mark >= 50) return 6;
```

```
    else if (mark >= 40) return 5;
```

```
    else return 0;
```

```
public class StudentSGPACalculator {
    public static void main (String [] args) {
        Scanner sc = new Scanner (System.in);
        System.out.print ("Enter the no of Subjects:");
        int numSubjects = sc.nextInt ();
        sc.nextLine ();
    }
}
```

```
Student student = new Student (numSubjects);
student.acceptDetails ();
student.displayDetails ();
```

Sc.close ();

}

### Output:-

Enter the number of Subjects: 3

Enter USN: 1bm23cs027

Enter name: Ameena

Enter credits and marks for each subject:

Credits for subject 1: 4

Marks for subject 1: 90

Credits for subject 2: 4

Marks for subject 2: 83

Credits for subject 3: 3

Marks for subject 3: 91

USN: 1bm23cs027

Name: Ameena

Subject 1 - credits: 4; Marks: 90

Subject 2 - Credits: 4, Marks: 83

Subject 3 - Credits: 3, Marks: 91

SGPA: 9.64

| 6 | 10 | 24 |

## Source Code

```
import java.util.Scanner;

class Student {

    String usn;
    String name;
    int[] credits;
    int[] marks;

    public Student(int numSubjects) {
        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 and marks for each subject:");
        for (int i = 0; i < credits.length; i++) {
```

```
        System.out.print("Credits for subject " + (i + 1) + ": ");
        credits[i] = sc.nextInt();
        System.out.print("Marks for subject " + (i + 1) + ": ");
        marks[i] = sc.nextInt();
    }

}

void displayDetails() {
    System.out.println("USN: " + usn);
    System.out.println("Name: " + name);

    for (int i = 0; i < credits.length; i++) {
        System.out.println("Subject " + (i + 1) + " - Credits: " +
credits[i] + ", Marks: " + marks[i]);
    }

    System.out.printf("SGPA: %.2f%n", calculateSGPA());
}

double calculateSGPA() {
    double totalCredits = 0;
    double totalPoints = 0;

    for (int i = 0; i < credits.length; i++) {
        double gradePoint = calculateGradePoint(marks[i]);
        totalPoints += gradePoint * credits[i];
    }
}
```

```
        totalCredits += credits[i];

    }

    return totalCredits > 0 ? totalPoints / totalCredits : 0;
}

double calculateGradePoint(int mark) {

    if (mark >= 90) return 10;

    else if (mark >= 80) return 9;

    else if (mark >= 70) return 8;

    else if (mark >= 60) return 7;

    else if (mark >= 50) return 6;

    else if (mark >= 40) return 5;

    else return 0;
}

public class StudentSGPACalculator {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

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

        int numSubjects = sc.nextInt();

        sc.nextLine();

        Student student = new Student(numSubjects);

        student.acceptDetails();
```

```
        student.displayDetails();
        sc.close();
    }
}
```

### **Output:**

```
D:\>javac StudentSGPACalculator.java

D:\>java StudentSGPACalculator
Enter the number of subjects: 3
Enter USN: 1bm23cs027
Enter Name: Ameena
Enter credits and marks for each subject:
Credits for subject 1: 4
Marks for subject 1: 90
Credits for subject 2: 4
Marks for subject 2: 83
Credits for subject 3: 3
Marks for subject 3: 91
USN: 1bm23cs027
Name: Ameena
Subject 1 - Credits: 4, Marks: 90
Subject 2 - Credits: 4, Marks: 83
Subject 3 - Credits: 3, Marks: 91
SGPA: 9.64
```

<https://github.com/Ameena1BM23CS27/java-lab/tree/main/java%20lab%203>

## Program 3

### Create Objects For Book

LAB - 3.



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

```
import java.util.Scanner;  
  
class Book {  
    private String name;  
    private String author;  
    private double price;  
    private int numPages;  
  
    public Book(String name, String author, double price,  
               int 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(double price){  
    this.price = price;  
}
```

```
public void setNumPages(int numPages){  
    this.numPages = numPages;  
}
```

```
public String getName(){  
    return name;  
}
```

```
public String getAuthor(){  
    return Author;  
}
```

```
public double getPrice(){  
    return price;  
}
```

```
public int getNumPages(){  
    return numPages;  
}
```

### ② Overrided

```
public String toString(){  
    return "Book Name:" + name + "\nAuthor:" +  
    + author + "\nPrice: $" + price + "\nNumber of pages"  
    + numPages;
```

}

Public class Main {

    public static void main (String[] args) {

        Scanner scanner = new

        Scanner (System.in);

        System.out.println ("Enter the number of books:");

        int n = scanner.nextInt();

        Scanner.nextLine();

        Book[] books = new Book[n];

        for (int i = 0; i < n; i++) {

            System.out.println ("Enter the details of the  
            book " + (i + 1));

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

            String name = scanner.nextLine();

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

            String author = scanner.nextLine();

            System.out.print ("Enter price: \$");

            double price = scanner.nextDouble();

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

            int numPages = scanner.nextInt();

            scanner.nextLine();

        books[i] = new Book(name, author, price, numPages)

```
System.out.println("n Book Details:");
for (int i=0; i<n; i++) {
```

```
    System.out.println("n Book "+(i+1));
    System.out.println(books[i].toString());
```

```
}
```

Output:-

Enter details for Book 1

Enter book name : It ends with us

Enter author's name : Collen Hoover

Enter price : \$ 8

Enter number of pages : 300

Enter details for Book 2 .

Enter book name : It starts with us

Enter author's name : Collen Hoover

Enter price : \$ 9

Enter number of pages : 320

Enter details for Book 3

Enter book name : Hunting Adeline

Enter author's name : H. D caulton

Enter price : \$ 10

Enter number of pages : 480



### Book Details:

#### Book 1

Book Name : It ends with us

Author : Collen Hoover

Price : \$8.0

Number of pages: 300

#### Book 2

Book name : It starts with us

Author : collen Hoover

price : \$9.0

Number of pages : 320

#### Book 3

Book name : Hunting Adeline

Author : H D Carlton

Price : \$10.0

Number of Pages : 480

23/10/24

## Sorce Code:

```
import java.util.Scanner;

class Book

{
    String name;
    String author;
    int price;
    int numPages;

    Book(String name, String author, int price, int numPages) {
        this.name = name;
        this.author = author;
        this.price = price;
        this.numPages = numPages;
    }

    @Override
    public String toString() {
        String bookDetails = "Book name: " + this.name + "\n" +
                            "Author name: " + this.author + "\n" +
                            "Price: " + this.price + "\n" +
                            "Number of pages: " + this.numPages + "\n";
        return bookDetails;
    }
}
```

```
public class Main {  
    public static void main(String[] args) {  
        Scanner s = new Scanner(System.in);  
  
        System.out.print("Enter the number of books: ");  
        int n = s.nextInt();  
  
        Book[] books = new Book[n];  
  
        for (int i = 0; i < n; i++) {  
            System.out.print("Enter name of book " + (i + 1) + ": ");  
            String name = s.next();  
            System.out.print("Enter author of book " + (i + 1) + ": ");  
            String author = s.next();  
            System.out.print("Enter price of book " + (i + 1) + ": ");  
            int price = s.nextInt();  
            System.out.print("Enter number of pages in book " + (i + 1) +  
": ");  
            int numPages = s.nextInt();  
  
            books[i] = new Book(name, author, price, numPages);  
        }  
  
        System.out.println("\nBook Details:");
```

```
        for (Book book : books) {  
            System.out.println(book);  
        }  
    s.close();  
    System.out.println("Ameena 1BM23CS027");  
}  
}
```

## Output:

```
D:\>java Main  
Enter the number of books: 2  
Enter name of book 1: it_ends_with_us  
Enter author of book 1: Collen_hoover  
Enter price of book 1: 500  
Enter number of pages in book 1: 320  
Enter name of book 2: Until_i_get_you  
Enter author of book 2: Clair_contraira  
Enter price of book 2: 150  
Enter number of pages in book 2: 470  
  
Book Details:  
Book name: it_ends_with_us  
Author name: Collen_hoover  
Price: 500  
Number of pages: 320  
  
Book name: Until_i_get_you  
Author name: Clair_contraira  
Price: 150  
Number of pages: 470  
  
Ameena 1BM23CS027
```

<https://github.com/Ameena1BM23CS27/java-lab/tree/main/java%20lab%204>

## Program 4

### Implement Abstract Class

Lab- Program-4

Date \_\_\_\_\_  
Page \_\_\_\_\_

Develop a lab 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.

```
Import .java.util.Scanner;  
class InputScanner{  
  
    Scanner sc = new Scanner(System.in);  
  
    }  
  
abstract class Shape extends InputScanner{  
    double dim1;  
    double dim2;  
    abstract double printArea();  
  
}  
  
class Rectangle extends Shape{  
  
    Rectangle(){  
        System.out.println("Enter the dimensions  
of the Rectangle:");  
  
        super.dim1 = sc.nextInt();  
        super.dim2 = sc.nextInt();  
    }  
  
    double printArea(){  
        System.out.println("Area of rectangle:");  
        return (dim1 * dim2);  
    }  
}
```

}

class Triangle extends Shape {

Triangle () {

System.out.println("Enter the dimensions of the  
triangle");

super.dim1 = sc.nextInt();

super.dim2 = sc.nextInt();

}

double printArea () {

System.out.println("Area of Triangle:");

return 0.5 \* dim1 \* dim2;

}

}

Circle () {

System.out.println("Enter the dimension(radius)  
of the circle:");

super.dim1 = sc.nextInt();

}

double printArea () {

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

return 3.14 \* dim1 \* dim1;

}

}

```
class AbstractDemo{  
    public static void main(String args[]){  
        Rectangle r1 = new Rectangle();  
        Triangle t = new Triangle();  
        Circle c = new Circle();
```

shape figref;

figref = r1

```
System.out.println("Area is:" + figref.printArea()  
+ "\n");
```

figref = t

```
System.out.println("Area is:" + figref.printArea()  
+ "\n");
```

figref = c;

```
System.out.println("Area is:" + figref.printArea() +  
"\n");
```

{

}

Output:-

Enter the dimensions of the Rectangle:

10

12

Enter the dimensions of the triangle:

7

8

Enter the dimension (radius) of the circle:

22

Area of rectangle:

Area is: 120.0

Area of Triangle:

Area is: 28.0

Area of Circle:

Area is: 1519.76.

RS

23/10/24

## Source Code:

```
import java.util.Scanner;

abstract class Shape {

    int dim1;

    int dim2;

    public Shape() {

        this.dim1 = 0;

        this.dim2 = 0;

    }

    public Shape(int dim1, int dim2) {

        this.dim1 = dim1;

        this.dim2 = dim2;

    }

    public abstract void printArea();

}

class Rectangle extends Shape {

    public Rectangle(int length, int width) {

        dim1 = length;

        dim2 = width;

    }

    public void printArea() {

        int area = dim1 * dim2;

    }

}
```

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

class Triangle extends Shape {
    public Triangle(int base, int height) {
        dim1 = base;
        dim2 = height;
    }

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

class Circle extends Shape {
    public Circle(int radius) {
        dim1 = radius;
        dim2 = 0;
    }

    public void printArea() {
        double area = Math.PI * dim1 * dim1;
        System.out.println("Area of Circle: " + area);
    }
}
```

```
    }

}

public class shapes
{
    public static void main(String[] args)
    {
        Scanner in = new Scanner(System.in);
        System.out.println("Enter length and width for Rectangle:");

        int length = in.nextInt();
        int width = in.nextInt();
        Shape rectangle = new Rectangle(length, width);
        rectangle.printArea();

        System.out.println("Enter base and height for Triangle:");

        int base = in.nextInt();
        int height = in.nextInt();
        Shape triangle = new Triangle(base, height);
        triangle.printArea();

        System.out.println("Enter radius for Circle:");

        int radius = in.nextInt();
        Shape circle = new Circle(radius);
```

```
        circle.printArea();

        in.close();

        System.out.println("Ameena 1BM23CS027");

    }

}
```

## **Output:**

```
D:\>javac shapes.java

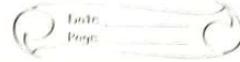
D:\>java shapes
Enter length and width for Rectangle:
10
12
Area of Rectangle: 120
Enter base and height for Triangle:
7
8
Area of Triangle: 28.0
Enter radius for Circle:
22
Area of Circle: 1520.5308443374597
Ameena 1BM23CS027
```

<https://github.com/Ameena1BM23CS27/java-lab/tree/main/java%205th%20laba>

## Program 5

### Bank Account Management

Lab 5



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 initial  
Balance, double minimumBalance, double  
serviceCharge)
```

{

```
super(customerName, accountNumber,  
initial Balance);
```

```
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 servicecharge:  
        " + serviceCharge);
```

```
        System.out.println ("Balance after deduction  
        " + balance);
```

```
}
```

```
}
```

```
public class Banks
```

```
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 acc no.");
```

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

```
    System.out.println ("Enter initial Balance.");
```

```
    double balance = sc.nextDouble();
```

## Output:-

Date \_\_\_\_\_  
Page \_\_\_\_\_

Enter customer name:

Shahrukh

Enter account number:

395376

Enter initial balance:

7000

Enter minimum balance:

500

Enter interest rate:

5

Enter service charge:

10

Enter choice:

1. Current Account

2. Savings Account

1

Customer Name: shahrukh

Account Number: 395376

Account is of current type

Enter choice:

1. Deposit

2. Withdraw

3. Display Balance

4. Exit

1

Enter amount to deposit:

3000

Deposited: 3000.0

Rs

## Source 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("Withdrew: " + 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 the minimum!");  
            balance -= serviceCharge;  
            System.out.println("Deducted service charge: " +  
                serviceCharge);  
            System.out.println("Balance after deduction: " + 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();
        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:\n1. Current Account\n2. Savings Account");
        int ch = sc.nextInt();
        System.out.println("Customer Name: " + name);
```

```
System.out.println("Account Number: " + acc_no);

switch (ch) {
    case 1: // Current Account
        System.out.println("Account is of Current type.");
        CurrentAccount ca = new CurrentAccount(name, acc_no,
balance, minimum_balance, service_charge);
        while (true) {
            System.out.println("Enter choice:\n1. Deposit\n2.
Withdraw\n3. Display Balance\n4. Exit");
            int c = sc.nextInt();
            if (c == 1) {
                System.out.println("Enter amount to deposit:");
                double amt = sc.nextDouble();
                ca.deposit(amt);
            } else if (c == 2) {
                System.out.println("Enter amount to withdraw:");
                double amt = sc.nextDouble();
                ca.withdraw(amt);
            } else if (c == 3) {
                ca.checkMinimumBalance();
                ca.displayBalance();
            } else {
                System.out.println("Exiting Current Account...");
```

```
        break;
```

```
    }
```

```
}
```

```
break;
```

```
case 2: // Savings Account
```

```
    System.out.println("Account is of Savings type.");
```

```
    SavingsAccount sa = new SavingsAccount(name, acc_no,  
balance, interest_rate);
```

```
    while (true) {
```

```
        System.out.println("Enter choice:\n1. Deposit\n2.  
Withdraw\n3. Display Balance\n4. Exit");
```

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

```
        if (c == 1) {
```

```
            System.out.println("Enter amount to deposit:");
```

```
            double amt = sc.nextDouble();
```

```
            sa.deposit(amt);
```

```
        } else if (c == 2) {
```

```
            System.out.println("Enter amount to withdraw:");
```

```
            double amt = sc.nextDouble();
```

```
            sa.withdraw(amt);
```

```
        } else if (c == 3) {
```

```
            sa.computeAndDepositInterest();
```

```
            sa.displayBalance();
```

```
    } else {
        System.out.println("Exiting Savings Account...");
        break;
    }
}

break;

default:
    System.out.println("Invalid choice! Exiting...");
}

System.out.println("Name: Ameena Yasmeen\nUSN:
1BM23CS027");
sc.close();
}

}
```

## **Output:**

```
Microsoft Windows [Version 10.0.19045.5131]
(c) Microsoft Corporation. All rights reserved.

D:\java 5th laba>javac Bank.java

D:\java 5th laba>java Bank.java
Enter customer name:
Shahrukh
Enter account number:
325376
Enter initial balance:
7000
Enter minimum balance:
500
Enter interest rate:
5
Enter service charge:
10
Enter choice:
1. Current Account
2. Savings Account
1
Customer Name: Shahrukh
Account Number: 325376
Account is of Current type.
Enter choice:
1. Deposit
2. Withdraw
3. Display Balance
4. Exit
1
Enter amount to deposit:
3000
Deposited: 3000.0
Enter choice:
1. Deposit
2. Withdraw
3. Display Balance
4. Exit
3
Current Balance: 10000.0
Enter choice:
1. Deposit
2. Withdraw
3. Display Balance
4. Exit
2
Enter amount to withdraw:
2000
Withdrew: 2000.0 | Balance is: 8000.0
```

<https://github.com/Ameena1BM23CS27/java-lab/tree/main/java%20lab%206>

## Program 6

### Implement Packages

Lab 6

Date \_\_\_\_\_  
Page \_\_\_\_\_

CIE / student.java  
java

```
Package CIE;  
import java.util.Scanner;
```

```
public class Student
```

```
{  
    protected String usn;  
    protected String name;  
    protected int sem;
```

```
    public void inputStudentDetails () {
```

```
        Scanner s = new Scanner (System.in)  
        System.out.println ("Enter USN: ");  
        usn = s.nextLine ();  
        System.out.print ("Enter name: ");  
        name = s.nextLine ();  
        System.out.print ("Enter semester: ");  
        sem = s.nextInt ();
```

```
}
```

```
    public void displayStudentDetails () {
```

```
        System.out.println ("USN: " + usn);  
        System.out.println ("Name: " + name);  
        System.out.println ("Semester: " + sem);
```

```
}
```

## 2. CIE / Internals

Package CIE;

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

```
public class Internals extends Student
```

```
{
```

```
protected int [] internalMarks = new int [5];
```

```
public void InputCIEmarks () {
```

```
Scanner s = new Scanner (System. in)
```

```
System.out.println ("Enter Internal Marks for
```

```
5 subjects :");
```

```
for (int i = 0; i < 5; i++)
```

```
{
```

```
System.out.print ("Subject " + (i + 1) + ":")
```

```
internalMarks [i] = s.nextInt ();
```

```
}
```

```
}
```

```
} // end of InputCIEmarks
```

```
/*
```

## 3. SEE / Externals.java

Package SEE;

```
import CIE.internals;
```

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

```
public class Externals extends Internals
```

```
{
```

```
private int[] seeMarks = new int[5];  
private int[] finalMarks = new int[5];
```

```
public void inputSEEmarks()
```

```
{
```

```
Scanner s = new Scanner(System.in);  
System.out.print("Enter SEE marks for  
5 subjects:");
```

```
for (int i = 0; i < 5; i++)
```

```
{
```

```
System.out.print("System" +  
(i + 1) + ":" + );
```

```
seeMarks[i] = s.nextInt();
```

```
}
```

```
public void calculateFinalMarks()
```

```
for (int i = 0; i < 5; i++)
```

```
{
```

```
finalMarks[i] = internalMarks[i] +  
seeMarks[i];
```

```
.
```

```
public void displayFinalMarks()
```

```
displayStudentDetails();
```

```
System.out.println("Final Marks
```

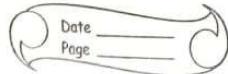
```
for (5 subjects:);
```

```
for (int i = 0; i < 5; i++)
```

```
{
```

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

Lab 7



## Creating your own exception subclasses

Write a program that demonstrates handling of exception

}  
}  
}.

### b. Main.java

```
import SEE.Externals;  
import java.util.Scanner;  
  
class Main {  
    public static void main (String [] args)  
    {  
        Scanner s = new Scanner (System.in);  
        System.out.print ("Enter no of students:");  
        int n = s.nextInt();
```

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

```
for (int i = 0; i < n; i++) {  
    System.out.println ("\n Enter details  
    for student " + (i + 1) + ":");  
    students [i] = new Externals ();  
    students [i].inputStudentDetails ();  
    students [i].inputFinalMarks ();  
    students [i].calculateFinalMarks ();  
}
```

System.out.println ("Final Marks of students");

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

System.out.println ("Student " + (i+1) + ":";  
students[i].displayFinalMarks();

}

}

Output:

Enter number of students: 1

Enter details for student 1:

Enter USN : IBM23CS027

Enter Name : Zehra Ameena Youmeen.

Enter Semester : 3

Enter Internal Marks for 5 subjects:

Subject 1 : 35

Subject 2 : 40

Subject 3 : 36

Subject 4 : 40

Subject 5 : 39.

Enter SEE Marks for 5 subjects:

Subject 1 : 97

Subject 2 : 98

Subject 3 : 91

Subject 4 : 94

Subject 5 : 95

## Final Marks of students:

Student :

USN : IBN03CS097

Name : Ameena Yasmeen

Semester : 3

Final Marks of 5 subjects:

Subject 1 : 81

Subject 2 : 85

Subject 3 : 89

Subject 4 : 89

Subject 5 : 91

~~RS~~

~~20/11/24~~

## **Source Code:**

```
package CIE;

import java.util.Scanner;

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

    public void inputStudentDetails() {
        Scanner s = new Scanner(System.in);
        System.out.print("Enter USN: ");
        usn = s.nextLine();
        System.out.print("Enter Name: ");
        name = s.nextLine();
        System.out.print("Enter Semester: ");
        sem = s.nextInt();
    }

    public void displayStudentDetails() {
        System.out.println("USN: " + usn);
        System.out.println("Name: " + name);
        System.out.println("Semester: " + sem);
    }
}
```

```
}

Package CIE;

import java.util.Scanner;

public class Internals extends Student {
    protected int[] internalMarks = new int[5];

    public void inputCIEmarks() {
        Scanner s = new Scanner(System.in);
        System.out.println("Enter Internal Marks for 5 subjects:");
        for (int i = 0; i < 5; i++) {
            System.out.print("Subject " + (i + 1) + ": ");
            internalMarks[i] = s.nextInt();
        }
    }
}

package SEE;

import CIE.Internals;

import java.util.Scanner;

public class Externals extends Internals {
    private int[] seeMarks = new int[5];
    private int[] finalMarks = new int[5];
```

```
public void inputSEEmarks() {  
    Scanner s = new Scanner(System.in);  
    System.out.println("Enter SEE Marks for 5 subjects:");  
    for (int i = 0; i < 5; i++) {  
        System.out.print("Subject " + (i + 1) + ": ");  
        seeMarks[i] = s.nextInt();  
    }  
}
```

```
public void calculateFinalMarks() {  
    for (int i = 0; i < 5; i++) {  
        finalMarks[i] = internalMarks[i] + seeMarks[i];  
    }  
}
```

```
public void displayFinalMarks() {  
    displayStudentDetails();  
    System.out.println("Final Marks for 5 subjects:");  
    for (int i = 0; i < 5; i++) {  
        System.out.println("Subject " + (i + 1) + ": " + finalMarks[i]);  
    }  
}  
import SEE.Externals;
```

```
import java.util.Scanner;

class Main {
    public static void main(String[] args) {
        Scanner s = new Scanner(System.in);
        System.out.print("Enter number of students: ");
        int n = s.nextInt();

        Externals[] students = new Externals[n];

        for (int i = 0; i < n; i++) {
            System.out.println("\nEnter details for student " + (i + 1) +
":");
            students[i] = new Externals();
            students[i].inputStudentDetails();
            students[i].inputCIEmarks();
            students[i].inputSEEmarks();
            students[i].calculateFinalMarks();
        }

        System.out.println("\nFinal Marks of Students:");
        for (int i = 0; i < n; i++) {
            System.out.println("\nStudent " + (i + 1) + ":");
            students[i].displayFinalMarks();
        }
    }
}
```

```

    }

    System.out.println("Name:Ameena
Yasmeen\nUSN:1BM23CS027");

}

}

```

## Output:

```

Microsoft Windows [version 10.0.19045.5011]
(c) Microsoft Corporation. All rights reserved.

C:\Users\DELL\Desktop\lab 5 java>javac -d . CIE/Student.java

C:\Users\DELL\Desktop\lab 5 java>javac -d . CIE/Internals.java

C:\Users\DELL\Desktop\lab 5 java>javac -d . SEE/Externals.java
error: file not found: SEE\Externals.java
Usage: javac <options> <source files>
use --help for a list of possible options

C:\Users\DELL\Desktop\lab 5 java>javac -d . SEE/Externals.java

C:\Users\DELL\Desktop\lab 5 java>javac Main.java

C:\Users\DELL\Desktop\lab 5 java>java Main.java
Enter number of students: 3

Enter details for student 1:
Enter USN: Exception in thread "main" java.util.NoSuchElementException: No line found
          at java.base/java.util.Scanner.nextLine(Scanner.java:1677)
          at CIE.Student.inputStudentDetails(Student.java:12)
          at Main.main(Main.java:15)
^C
C:\Users\DELL\Desktop\lab 5 java>javac Main.java

C:\Users\DELL\Desktop\lab 5 java>java Main.java
Enter number of students: 3

Enter details for student 1:
Enter USN: 27
Enter Name: Ameena
Enter Semester: 3
Enter Internal Marks for 5 subjects:
Subject 1: 35
Subject 2: 39
Subject 3: 33
Subject 4: 35
Subject 5: 33
Enter SEE Marks for 5 subjects:
Subject 1: 85
Subject 2: 91
Subject 3: 94
Subject 4: 99
Subject 5: 93

```

```
Enter details for student 2:  
Enter USN: 21  
Enter Name: shahrukh  
Enter Semester: 2  
Enter Internal Marks for 5 subjects:  
Subject 1: 33  
Subject 2: 34  
Subject 3: 33  
Subject 4: 34  
Subject 5: 34  
Enter SEE Marks for 5 subjects:  
Subject 1: 88  
Subject 2: 89  
Subject 3: 87  
Subject 4: 88  
Subject 5: 87  
  
Enter details for student 3:  
Enter USN: 12  
Enter Name: syed  
Enter Semester: 3  
Enter Internal Marks for 5 subjects:  
Subject 1: 34  
Subject 2: 34  
Subject 3: 32  
Subject 4: 32  
Subject 5: 31  
Enter SEE Marks for 5 subjects:  
Subject 1: 88  
Subject 2: 89  
Subject 3: 86  
Subject 4: 95  
Subject 5: 99
```

```
Final Marks of Students:  
  
Student 1:  
USN: 27  
Name: Ameena  
Semester: 3  
Final Marks for 5 subjects:  
Subject 1: 120  
Subject 2: 130  
Subject 3: 127  
Subject 4: 134  
Subject 5: 126  
  
Student 2:  
USN: 21  
Name: shahrukh  
Semester: 2  
Final Marks for 5 subjects:  
Subject 1: 121  
Subject 2: 123  
Subject 3: 120  
Subject 4: 122  
Subject 5: 121  
  
Student 3:  
USN: 12  
Name: syed  
Semester: 3  
Final Marks for 5 subjects:  
Subject 1: 122  
Subject 2: 123  
Subject 3: 118  
Subject 4: 127  
Subject 5: 130  
Name: Ameena Yasmeen  
USN: 1BM23CS027  
  
C:\Users\DELL\Desktop\lab 5.java>
```

<https://github.com/Ameena1BM23CS27/java-lab/tree/main/java%20lab%207>

## Program 7

### Implement Exception Handling

Lab - 7.

Date \_\_\_\_\_  
Page \_\_\_\_\_

Q) Write a program to demonstrate handling of exception in inheritance tree

```
import java.util.Scanner;
class WrongAgeException extends Exception {
    public WrongAgeException (String message)
        super (message);
}

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

class Father {
    private int age;
    public Father (int age) throws WrongAgeException {
        if (age < 0) {
            throw new WrongAgeException ("Wrong Age");
        }
        this.age = age;
    }
    public int getAge () {
        return age;
    }
}
```

class son extends father {  
    private & int sonAge;  
    public Son (int fatherAge, int sonAge)  
        throws WrongAgeException, SonAgeException  
    {

        super(fatherAge);

    } (sonAge >= fatherAge) {

        throws new SonAgeException ("son's age cannot be greater than or equal  
        to father's age");

    } this.sonAge = sonAge

    } public int getsonAge ()

    { return sonAge;

    }

public class FatherSon {

    public static void main (String [] args)

    { while (true) {

        Scanner sc = new Scanner (System.

        System.out.println ("Enter father's age");  
        int fatherAge = sc.nextInt ();

        System.out.print ("Enter son's Age: ");

        int sonAge = sc.nextInt ();

        try

        { Son son = new Son (fatherAge, sonAge);

            System.out.println ("Accepted  
            successfully");

```

    }
    catch (WrongAgeException e) {
        System.out.println(e.getMessage());
    }
}

```

```

    }
    catch (SonAgeException e) {
        System.out.println(e.getMessage());
    }
}

```

```

System.out.println("Would you like to re-enter
details (y/n);");

```

```

String input = sc.nextLine();

```

```

if (input.equalsIgnoreCase("n")) {
    break;
}

```

```

}
}
}

```

Output:-

Enter father's Age: 41

Enter Son's Age: 70

Son's Age cannot be greater than or equal to father's age

Would you like to re-enter details (y/n)

y

Enter Father's Age: 41

Enter Son's Age: 21

Accepted Successfully

Would you like to re-enter details (y/n)

82  
20/11/24

## **Source Code:**

```
import java.util.Scanner;

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

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

class Father {
    private int age;
    public Father(int age) throws WrongAgeException {
        if (age < 0) {
            throw new WrongAgeException("Wrong age");
        }
        this.age = age;
    }
    public int getAge() {
        return age;
    }
}
```

```
    }

}

class Son extends Father {

    private int sonAge;

    public Son(int fatherAge, int sonAge) throws WrongAgeException,
SonAgeException {

        super(fatherAge);

        if (sonAge >= fatherAge) {

            throw new SonAgeException("Son's age cannot be greater
than or equal to father's age");

        }

        this.sonAge = sonAge;

    }

    public int getSonAge() {

        return sonAge;

    }

}

public class FatherandSon{

    public static void main(String[] args) {

        while(true){

            Scanner sc = new Scanner(System.in);

            System.out.print("Enter Father's Age: ");

            int fatherAge = sc.nextInt();


```

```
System.out.print("Enter Son's Age: ");
int sonAge = sc.nextInt();
try {
    Son son = new Son(fatherAge, sonAge);
    System.out.println("Accepted Succesfully");
}
catch (WrongAgeException e) {
    System.out.println(e.getMessage());
}
catch (SonAgeException e) {
    System.out.println(e.getMessage());
}
System.out.println("Would you like to re-enter details (Y/n)");
String input = sc.next();
if (input.equalsIgnoreCase("n")) {
    break;
}
}
}
```

## Output:

```
D:\>javac quad.java

D:\>java quad
Enter coefficient of a:
1
Enter coefficient of b:
5
Enter coefficient of c:
2
The roots are real and different
Root 1:-0.4384471871911697
Root 2:-4.561552812808831
Ameena Yasmeen 1BM23CS027

D:\>javac quad.java

D:\>java quad
Enter coefficient of a:
1
Enter coefficient of b:
9
Enter coefficient of c:
6
The roots are real and different
Root 1:-0.7250827823646251
Root 2:-8.274917217635375
Ameena Yasmeen 1BM23CS027

D:\>javac quad.java

D:\>java quad
Enter coefficient of a:
4
Enter coefficient of b:
5
Enter coefficient of c:
6
The roots are complex
Root 1:-0.625+1.0532687216470449i
Root 2:-0.625-1.0532687216470449i
Ameena Yasmeen 1BM23CS027
```

<https://github.com/Ameena1BM23CS27/java-lab/tree/main/java%20lab%208>

## Program 8

### MultiThreading, creating Threads in Java

Lab - 8

```
class Message 1 extends Thread {
    @Override
    public void run() {
        try {
            while(true) {
                System.out.println("BMS college of
Engineering");
                Thread.sleep(10000);
            }
        } catch(InterruptedException e) {
            System.out.println(e);
        }
    }
}

public class MultiThreadExample {

    class Message 2 implements Runnable {
        @Override
        public void run() {
            try {
                while(true) {
                    System.out.println("CSE")
                    Thread.sleep(2000);
                }
            } catch(InterruptedException e) {
                System.out.println(e);
            }
        }
    }
}
```

```
public class MultithreadExample {  
    public static void main (String [] args)  
    {  
        Message1 thread1 = new Message1 ();  
        thread1.start ();  
    }  
}
```

```
Message2 message2 = new Message2 ();  
Thread thread2 = new Thread (message2);  
thread2.start ();  
}
```

Output:-

BMS College of Engineering

CSE

CSE

CSE

CSE

CSE

*Ran*  
*5/12/24*

## **Source code:**

```
class Message1 extends Thread {  
    @Override  
    public void run() {  
        try {  
            while (true) {  
                System.out.println("BMS College of Engineering");  
                Thread.sleep(10000);  
            }  
        } catch (InterruptedException e) {  
            System.out.println(e);  
        }  
    }  
  
    class Message2 implements Runnable {  
        @Override  
        public void run() {  
            try {  
                while (true) {  
                    System.out.println("CSE");  
                    Thread.sleep(2000);  
                }  
            } catch (InterruptedException e) {  
                System.out.println(e);  
            }  
        }  
    }  
}
```

```
    }
}

}

public class MultiThreadExample {
    public static void main(String[] args) {
        System.out.println("Ameena Yasmeen\n1BM23CS027");
        Message1 thread1 = new Message1();
        thread1.start();

        Message2 message2 = new Message2();
        Thread thread2 = new Thread(message2);
        thread2.start();
    }
}
```

**Output:**

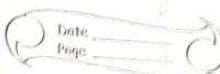
```
D:\java lab 8>java MultiThreadExample
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
```

<https://github.com/Ameena1BM23CS27/java-lab/tree/main/java%20lab%209>

## Program 9

### Interface To Perform Integer Division

Lab program - 9



Interface to perform integer division

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

```
class SwingDemo {  
    SwingDemo () {
```

```
        JFrame jfrm = new JFrame ("Divide App");  
        jfrm . setSize (275, 150);  
        jfrm . setLayout (new FlowLayout());
```

```
        jfrm . setDefaultCloseOperation (JFrame . EXIT_ON_CLOSE)
```

```
        JLabel jlab = new JLabel ("Enter the divisor and  
        dividend : ");
```

```
        JTextField aJF = new JTextField (8);  
        JTextField bJF = new JTextField (8);
```

```
        JButton button = new JButton ("Calculate");
```

```
        JLabel erkl = new JLabel ();  
        JLabel alab = new JLabel ();  
        JLabel blab = new JLabel ();  
        JLabel anslab = new JLabel ();
```

```
jfrm . add (erkl);  
jfrm . add (alab);  
jfrm . add (blab);  
jfrm . add (anslab);
```

```

jfrm.add(button);
jfrm.add(alab);
jfrm.add(blab);
jfrm.add(anslab);

```

```

button.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent evt) {
        try {
            int a = Integer.parseInt(qtf.getText());

```

```

            int b = Integer.parseInt(btf.getText());
            int ans = a / b;

```

```

            alab.setText("A = " + a);
            blab.setText("B = " + b);
            anslab.setText("Ans = " + ans);
            err.setText("");
        } catch (NumberFormatException e) {

```

```

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

```

```

            alab.setText("");
            blab.setText("");
            anslab.setText("");
            err.setText("B should be Nonzero!");
        }
    }
});
```

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

Output:-

Enter the divisor and dividend:

400 17

calculate A=400, B=17 Ans=23

~~Rs~~

~~512124~~

## Source Code:

```
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 dividend:");
        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);

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

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

```

        blab.setText("");
        anslab.setText("");
        err.setText("B should be NON zero!");
    }
}

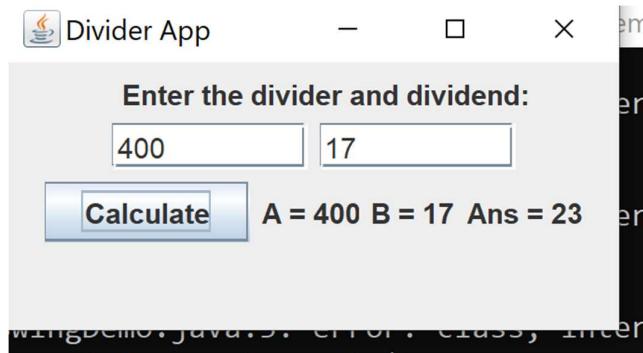
});

jfrm.setVisible(true);
}

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

```

## Output:



[https://github.com/Ameena1BM23CS27/java-lab/tree/main/java%20lab%2010%20\(a\)](https://github.com/Ameena1BM23CS27/java-lab/tree/main/java%20lab%2010%20(a))

## Program 10 (a)

### Implement Deadlock

Lab 10

Date \_\_\_\_\_  
Page \_\_\_\_\_

#### Implement Deadlock

```
class SharedResource {
    private int data;
    private boolean isDataAvailable = false;

    public synchronized void produceData(int data)
        throws InterruptedException {
        while (!isDataAvailable) {
            wait();
        }
        this.data = data;
        System.out.println("produced :" + data);
        isDataAvailable = true;
        notify();
    }

    public synchronized void consumeData() throws
        InterruptedException {
        while (!isDataAvailable) {
            wait();
        }
        System.out.println("consumed :" + data);
        isDataAvailable = false;
        notify();
    }

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

SharedResource sharedResource = new SharedResource();

Thread producer = new Thread(() -> {  
try {

for (int i=0; i<5; i++) {

sharedResource.produceData(i);

Thread.sleep(1000);

} catch (InterruptedException e) {

e.printStackTrace();

}

} );

Thread consumer = new Thread(() -> {

try {

for (int i=0; i<5; i++) {

sharedResource.consumeData();

Thread.sleep(1500);

}

} catch (InterruptedException e) {

e.printStackTrace();

}

} );

producer.start();

consumer.start();

}.

Output:-

Put:0

Get:0

Put:1

Get:1

Put:2

Get:2

Put:3

Get:3

~~Put~~

~~12/12/28~~

## Source Code:

```
class SharedResource {  
    private int data;  
    private boolean isDataAvailable = false;  
  
    public synchronized void produceData(int data) throws  
    InterruptedException {  
        while (isDataAvailable) {  
            wait();  
        }  
        this.data = data;  
        System.out.println("Put: " + data);  
        isDataAvailable = true;  
        notify();  
    }  
  
    public synchronized void consumeData() throws  
    InterruptedException {  
        while (!isDataAvailable) {  
            wait();  
        }  
        System.out.println("Got: " + data);  
        isDataAvailable = false;  
        notify();  
    }  
}
```

```
    }

}

public class IPCDemo {
    public static void main(String[] args) {
        // Print your name once at the beginning
        System.out.println("Ameena Yasmeen\n 1BM23CS027");

        SharedResource sharedResource = new SharedResource();

        Thread producer = new Thread(() -> {
            try {
                for (int i = 0; i < 5; i++) {
                    sharedResource.produceData(i);
                    Thread.sleep(1000);
                }
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        });

        Thread consumer = new Thread(() -> {
            try {
                for (int i = 0; i < 5; i++) {
```

```
        sharedResource.consumeData();

        Thread.sleep(1500);

    }

} catch (InterruptedException e) {
    e.printStackTrace();
}

});

producer.start();

consumer.start();

}
```

Output:

```
D:\java lab 10 (a)>java IPCDemo
Ameena Yasmeen
1BM23CS027
Put: 0
Got: 0
Put: 1
Got: 1
Put: 2
Got: 2
Put: 3
Got: 3
Put: 4
Got: 4
D:\java lab 10 (a)>
```

<https://github.com/Ameena1BM23CS27/java-lab/tree/main/java%20lab%2010%20b>

## Program 10(b)

### Implement Inter-Process Communication

Date \_\_\_\_\_  
Page \_\_\_\_\_  
Implement Inter-process Communication.

Source-code :

```
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");  
        }  
    }  
}
```

Synchronized (b)

{

```
    System.out.println(name + " trying to call  
    B.last());  
    b.last();  
}
```

};

void last()

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

}

class B{

Synchronized void B() {

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

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

try {

Thread.sleep(1000);

}

catch (Exception e) {

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

}

Synchronized (a) {

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

a.last()

}

void last()

System.out.println ("Inside B's lock");

}

class Deadlock implements Runnable {

A a = new A();

B b = new B();

Deadlock () {

System.out.println ("Ameena \Yasmeen");

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

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

a.foot (b);

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

}

public void run () {

b.boo (a);

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

}

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

```
}
```

Output:

Ameena Vaameen

Mainthread entered A.foo

Racingthread entered B.bar

## Source Code:

```
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");  
  
        }  
  
        synchronized (b) {
```

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

```
}

void last() {
    System.out.println("Inside B.last");
}

}

class Deadlock implements Runnable {

    A a = new A();
    B b = new B();

    Deadlock() {
        System.out.println("Ameena Yasmeen\n1BM23CS027");

        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:\java lab 10 b>java Deadlock
PAmeena Yasmeen
1BM23CS027
c
>MainThread entered A.foo
j>RacingThread entered B.bar
L
-----
-----
```