

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



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

*Submitted by*

Bramha Anilkumar Bajannavar(**1BM23CS071**)

*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 **Bramha Anilkumar Bajannavar (1BM23CS071)**, 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.

Swathi Sridharan Assistant Professor Department of CSE, BMSCE	Dr. Kavitha Sooda Professor & HOD Department of CSE, BMSCE
---	--

## Index

<b>Sl. No.</b>	<b>Date</b>	<b>Experiment Title</b>	<b>Page No.</b>
1	01/10/2024	Quadratic Equation	4-5
2	08/10/2024	Calculate SGPA of Students	6-10
3	15/10/2024	Book Details	11-16
4	22/10/2024	Abstract Classes-Animal and Shapes	17-25
5	29/10/2024	Bank Details	26-31
6	12/11/2024	Packages- Student Marks	31-38
7	19/11/2024	Interfaces- Shapes	39-42
8	26/11/2024	Exception Handling	43-46
9	03/12/2024	Threads	47-49
10	03/12/2024	Graphical User Interface	50-54

Github Link:

<https://github.com/BramhaBajannavar/OOJ-Lab>

## 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 solution

Algorithm:

```
8b Java program to use quadratic formula
import java.util.Scanner;
public class quadratic {
    public static void main (String [] args) {
        Scanner cin = new Scanner (System.in);
        System.out.println ("Enter 3 numbers");
        int a = cin.nextInt();
        int b = cin.nextInt();
        int c = cin.nextInt();
        float d = (b*b) - (4*a*c);
        if (d == 0) {
            System.out.println ("roots are equal");
            System.out.println ("roots are ");
            System.out.println (-b / (2*a));
        } else if (d > 0) {
            System.out.println ("roots are distinct");
            System.out.println ((-b + Math.sqrt(d)) / (2*a));
            System.out.println ((-b - Math.sqrt(d)) / (2*a));
        } else
            System.out.println ("roots are not real");
    }
}

Enter 3 numbers: 1 -2 1
Roots are equal
Roots are
1

```

Sur  
1/10/2021

Code:

```
import java.util.Scanner;
public class Quad{
    public static void main(String[] args){
        Scanner s=new Scanner(System.in);
        System.out.println("Enter the coefficients:");
        int a=s.nextInt();
        int b=s.nextInt();
        int c=s.nextInt();
        int d=b*b-4*a*c;
        if(d==0){
            System.out.println("Roots are equal");
            System.out.println("Roots are:");
            System.out.println(-b/2*a);
        }
        else if(d>0){
            System.out.println("Roots are unique");
            System.out.println((-b+Math.sqrt(d))/(2*a));
            System.out.println((-b-Math.sqrt(d))/(2*a));
        }
        else{
            System.out.println("No real roots");
        }
    }
}
```

OUTPUT:

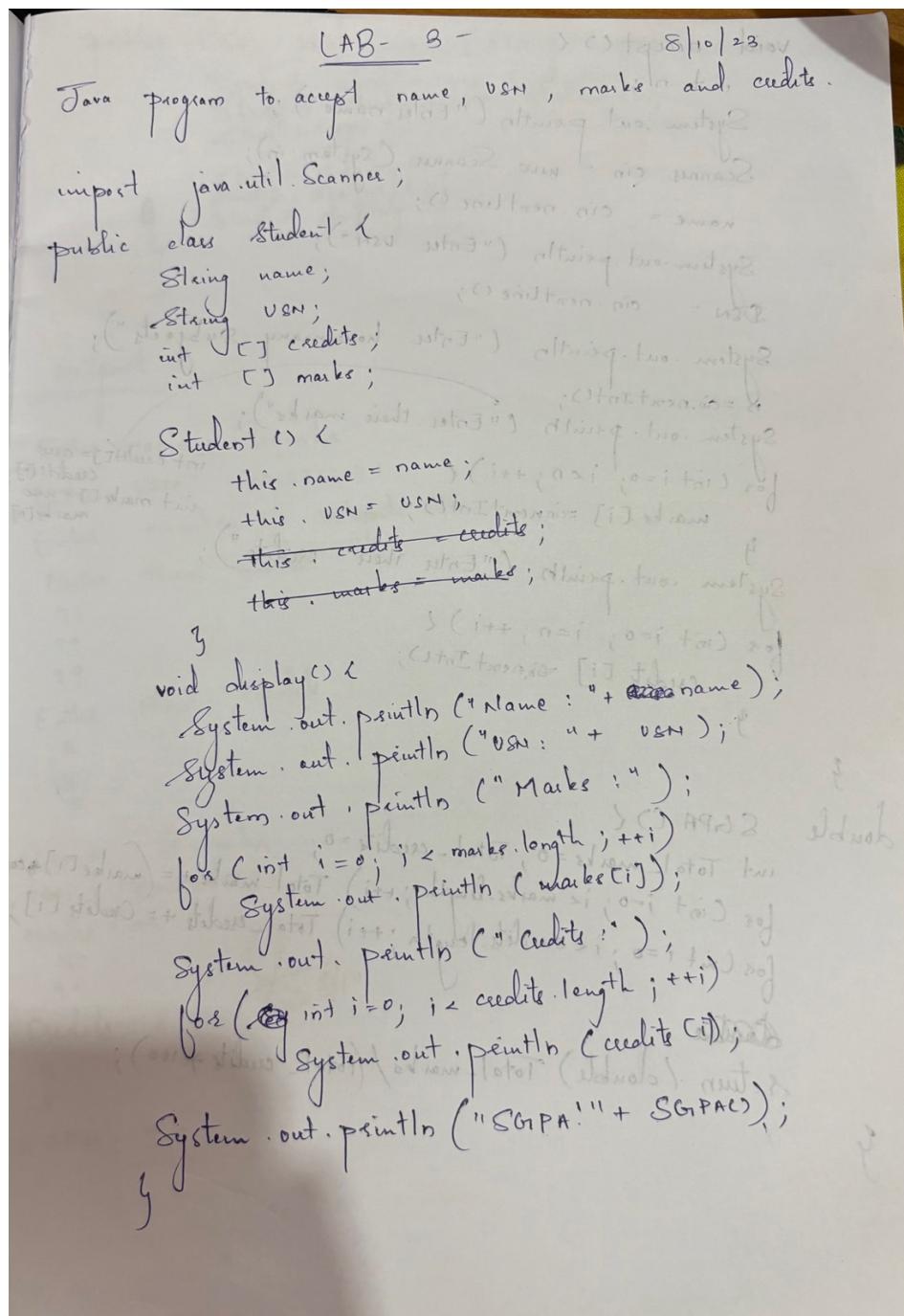
```
C:\Users\trish\OneDrive\Desktop\test>javac Quad.java

C:\Users\trish\OneDrive\Desktop\test>java Quad
Name : Bramha Anilkumar Bajannavar
USN : 1BM23CS071
Enter the coefficients:
1
-2
1
Roots are equal
Roots are:
1
```

## 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

Algorithm:



```

void accept() {
    int n;
    System.out.println("Enter name");
    Scanner cin = new Scanner(System.in);
    name = cin.nextLine();
    System.out.println("Enter USN");
    USN = cin.nextLine();
    System.out.println("Enter how many subjects");
    n = cin.nextInt();
    System.out.println("Enter their marks");
    for (int i=0; i<n; ++i) {
        marks[i] = cin.nextInt();
    }
    System.out.println("Enter their credits");
    for (int i=0; i<n; ++i) {
        credits[i] = cin.nextInt();
    }
}

double SGPA() {
    int Total_marks = 0, Total_credits = 0;
    for (int i=0; i<marks.length; ++i) Total_marks += marks[i];
    for (int i=0; i<credits.length; ++i) Total_credits += credits[i];
    return (double) Total_marks / (Total_credits * 100);
}

```

```
public static void main (String [] args) {  
    Student stud1 = new Student ();  
    Student stud2 = new Student ();  
    stud1.accept ();  
    stud1.display ();  
    stud2.accept ();  
    stud2.display ();  
}
```

Output

Enter name : Bramha  
Enter USN : 1BM23CS071  
Enter number of subjects : 3  
Enter their marks :

99

99

99

Enter their credits :

3

3

3

Name : Bramha

USN : 1BM23CS071

Marks : 99

99

99

Credits : 3

3

3

S G PA : 9.9

8/10

Code:

```
import java.util.Scanner;

public class Student {
    String name;
    String Usn;
    int n;
    int [] marks;
    int [] credits;

    Student(){
        this.name = "";
        this.Usn = "";
    }

    void display(){
        System.out.println("Name : " +name);
        System.out.println("USN : "+ Usn);
        System.out.println("Marks: ");
        for (int i=0; i< marks.length; ++i) System.out.println(marks[i]);
        for (int i=0; i< credits.length; ++i) System.out.println(credits[i]);
        System.out.println("SGPA : "+SGPA());
    }

    double SGPA() {
        int totalMarks = 0, totalCredits = 0;
        for (int i = 0; i < n; ++i) {
            totalMarks += marks[i] * credits[i];
            totalCredits += credits[i];
        }
        return (double) totalMarks / (totalCredits*10);
    }

    void accept(){
        System.out.println("Enter name");
        Scanner cin = new Scanner(System.in);
        name = cin.nextLine();
        System.out.println("Enter USN");
        Usn = cin.nextLine();
        System.out.println("Enter number of subjects");
        int n = cin.nextInt();
    }
}
```

```

        marks = new int[n];
        credits = new int[n];
        System.out.println("Enter their marks");
        for (int i=0;i<n;++i) marks[i] = cin.nextInt();
        System.out.println("enter their credits");
        for (int i=0; i< n;++i) credits[i] = cin.nextInt();
    }

public static void main(String[] args){
    System.out.println("Name: Bramha Anilkumar Bajannar");
    System.out.println("USN : 1BM23CS071");
    Student stud1 = new Student();
        Student stud2 = new Student();
    stud1.accept();
    stud1.display();
        stud2.accept();
    stud2.display();
}
}

```

OUTPUT:

```

C:\Users\trish\OneDrive\Desktop\test>java Student
Name: Bramha Anilkumar Bajannar
USN : 1BM23CS071
Enter name
Bramha
Enter USN
1BM23CS071
Enter number of subjects
2
Enter their marks
32
32
enter their credits
3
2
Name : Bramha
USN : 1BM23CS071
Marks:
32
32
3
2
SGPA : NaN
Enter name
Daiivya
Enter USN
1BM23CS070
Enter number of subjects
2
Enter their marks
32
32
enter their credits
3
3
Name : Daiivya
USN : 1BM23CS070
Marks:
32
32
3
3
SGPA : NaN
C:\Users\trish\OneDrive\Desktop\test>

```

### 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.

Algorithm:

LAB - 3 15/10/24

Program to create book class.

```
import java.util.Scanner;
public class Book {
    String name;
    String author;
    double price;
    int pages;
    String getname() {
        return name;
    }
    public Book(String name, String author, double price, int pages) {
        this.name = name;
        this.author = author;
        this.price = price;
        this.pages = pages;
    }
    public String getname() {
        return name;
    }
    public void setname() {
        this.name = name;
    }
}
```

```

public String getAuthor() {
    return author;
}

public double getPrice() {
    return price;
}

public int getPages() {
    return pages;
}

public void setAuthor() {
    this.author = author;
}

public void setPrice() {
    this.price = price;
}

public void setPages() {
    this.pages = pages;
}

@Override
public String toString() {
    System.out.println("Name:" + name + "Author:" + Author + "Price:" + Price
        + "No. of pages" + pages);
}

```

```

public static void main (String [] args) {
    Scanner cin = new Scanner (System .in);
    System.out.println ("Enter number of books");
    int n = cin.nextInt ();
    Book [ ] book = new Book [n];
    for (int i=0; i<n; ++i) {
        System.out.println ("Enter values of Book " + (i+1));
        System.out.println ("Enter book name");
        String name = cin.nextLine();
        System.out.println ("Enter author name");
        String author = cin.nextLine();
        System.out.println ("Enter number of pages");
        int pages = cin.nextInt();
        System.out.println ("Enter the price");
        double price = cin.nextDouble();
        book [i] = new Book (name, author, price, pages);
    }
}

```

~~System.out.println ("Details of each book are");~~  
~~for (int i=0; i<n; ++i) {~~  
~~book [i].toString();~~

see

Code:

```
import java.util.*;
```

```
public class Book{  
    String name;  
    String author;  
    double price;  
    int pages;
```

```
Book(String name,String author,double price,int pages){  
    this.name = name;  
    this.author = author;  
    this.price = price;  
    this.pages= pages;  
}
```

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

```
public void setname(){  
    this.name = name;  
}
```

```
public String getauthor(){  
    return author;  
}
```

```
public void setauthor(){  
    this.author = author;  
}
```

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

```
public void setprice(){  
    this.price = price;  
}
```

```

public int getpages(){
    return pages;
}

public void setpages(){
    this.pages = pages;
}

@Override
public String toString(){
    return "Name: " + name + " Author: " + author + " Price: " + price + " No of pages: "+ pages ;
}

public static void main(String [] args){
    System.out.println("Name: Bramha Anilkumar Bajannavar");
    System.out.println("USN: 1BM23CS071");
    Scanner cin = new Scanner(System.in);
    System.out.println("Enter the number of books");
    int n = cin.nextInt();
    cin.nextLine();

    Book[] book = new Book[n];
    System.out.println("Enter the values of each book");
    System.out.println("\n");
    for (int i=0;i<n;i++)
    {
        System.out.println("Enter the name of the book");
        String name=cin.nextLine();
        System.out.println("Enter the name of the author");
        String author=cin.nextLine();
        System.out.println("Enter the price");
        double price=cin.nextDouble();
        System.out.println("Enter the no of pages");
        int pages=cin.nextInt();
        book[i]=new Book(name,author,price,pages);
        cin.nextLine();
    }
}

```

```
System.out.println("Details of each book are");
for (int i=0; i<n;i++){
    System.out.println("Book "+(i+1)+book[i].toString());
}

}
}
```

OUTPUT:

```
C:\Users\trish\OneDrive\Desktop\test>javac Book.java
C:\Users\trish\OneDrive\Desktop\test>java Book
Name: Bramha Anilkumar Bajannavar
USN: 1BM23CS071
Enter the number of books
2
Enter the values of each book

Enter the name of the book
Harry Potter
Enter the name of the author
J K Rowling
Enter the price
399
Enter the no of pages
600
Enter the name of the book
ASOIAF
Enter the name of the author
George R R Martin
Enter the price
999
Enter the no of pages
900
Details of each book are
Book 1Name: Harry Potter Author: J K Rowling Price: 399.0 No of pages: 600
Book 2Name: ASOIAF Author: George R R Martin Price: 999.0 No of pages: 900
C:\Users\trish\OneDrive\Desktop\test>
```

#### Program 4:

Create an abstract class Animal with methods eat and sleep. Create 3 subclasses Lion, Tiger and deer that extends animal class. Implement these methods differently.

Algorithm:

LAB - Java abstraction 22/10/24

Create abstract class Animal with eat and sleep methods. Create subclasses lion, tiger and deer.

```
public abstract class Animal {  
    public abstract void eat();  
    public abstract void sleep();  
  
    public class Lion extends Animal {  
        @Override  
        public void eat() {  
            System.out.println("The lion is eating");  
        }  
  
        @Override  
        public void sleep() {  
            System.out.println("The lion is sleeping");  
        }  
  
    }  
  
    public class Tiger extends Animal {  
        @Override  
        public void eat() {  
            System.out.println("The tiger is eating");  
        }  
  
        @Override  
        public void sleep() {  
            System.out.println("The tiger is sleeping");  
        }  
    }  
}
```

public class Deer extends Animal { } }

@ override

public void eat() {  
System.out.println ("The Deer is sleeping");

@ override

public void sleep() {  
System.out.println ("The Deer is sleeping");

public static void main (String [] args) {

Animal lion = new Lion();

Animal tiger = new Tiger();

Animal deer = new Deer();

lion.eat();

lion.sleep();

tiger.eat();

tiger.sleep();

deer.eat();

deer.sleep();

Output

the lion is eating  
 the lion is sleeping  
 the tiger is eating  
 the tiger is sleeping  
 the deer is eating  
 the deer is sleep

OP seen  
gt  
1.0124

Code:

```

public abstract class Animal{
    public abstract void eat();
    public abstract void sleep();

    public static void main(String [] args){
        Animal lion = new Lion();
        Animal tiger = new Tiger();
        Animal deer = new Deer();

        lion.eat();
        lion.sleep();
        tiger.eat();
        tiger.sleep();
        deer.eat();
        deer.sleep();
    }
}
  
```

```
public class Lion extends Animal{
```

```
    @Override  
    public void eat(){  
        System.out.println("The lion is eating");  
    }
```

```
    @Override  
    public void sleep(){  
        System.out.println("The lion is sleeping");  
    }  
}
```

```
public class Tiger extends Animal{  
    @Override  
    public void eat(){  
        System.out.println("The tiger is eating");  
    }
```

```
    @Override  
    public void sleep(){  
        System.out.println("The tiger is sleeping");  
    }  
}
```

```
public class Deer extends Animal{  
    @Override  
    public void eat(){  
        System.out.println("The deer is eating");  
    }
```

```
    @Override  
    public void sleep(){  
        System.out.println("The deer is sleeping");  
    }
```

}

OUTPUT:

```
C:\Users\Admin\Documents\Bramha>java Animal
The lion is eating
The lion is sleeping
The tiger is eating
The tiger is sleeping
The deer is eating
The deer is sleeping
```

```
C:\Users\Admin\Documents\Bramha>
```

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

Algorithm:

~~Program to print area using abstract classes~~

~~Creating a base class~~

~~Shape~~

~~(A, S) are sub classes~~

~~int p, q;~~

~~public abstract void printarea();~~

~~Shape()~~

~~this.p = p; if (area) then setting lion sleeping~~

~~this.q = q; if (area) then setting tiger sleeping~~

~~Y~~

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

~~Shape rectangle = new Rectangle (4,5);~~

~~Shape triangle = new Triangle (3,4);~~

~~Shape circle = new Circle (5,0);~~

~~rectangle.printarea ();~~

~~triangle.printarea ();~~

~~circle.printarea ();~~

~~Y~~

public class Rectangle extends Shape {

System.out.println("Area of Rectangle is " +

Rectangle (int l, int b) {  
super (l, b);

@Override

public void paintarea () {

System.out.println ("Area of rectangle is " + (l \* b));

} public class Triangle extends Shape {

Triangle (int b, int h) {

super (b, h);

@Override

public void paintarea () {

System.out.println ("Area of triangle is " + 0.5 \* b \* h);

} public class Circle extends Shape {

Circle (int r) {

super (r, 0);

@Override

public void paintarea () {

System.out.println ("Circle area is " + (3.14 \* r \* r));

executed.

Q3  
Area  
Area  
Area  
Area  
seen  
gr  
not

Output

Area of rectangle is 25

Area of triangle is 5.0

Area of circle is 78.5

See

GT  
soln/2

Code:

```
public abstract class Shape {  
    int p,q;  
    public abstract void printarea();  
    Shape(int p,int q){  
        this.p =p;  
        this.q=q;  
    }  
    public static void main(String [] args){  
        Shape rectangle = new Rectangle(5,5);  
        Shape triangle = new Triangle(5,2);  
        Shape circle = new Circle(5);  
  
        rectangle.printarea();  
        triangle.printarea();
```

```

        circle.printarea();
    }
}

public class Rectangle extends Shape{
    Rectangle(int l , int b){
        super(l,b);
    }

    @Override
    public void printarea(){
        System.out.println("Area of rectangle is "+ p*q);
    }
}

public class Triangle extends Shape{
    Triangle(int l , int b){
        super(l,b);
    }

    @Override
    public void printarea(){
        System.out.println("Area of triangle is "+ 0.5*p*q);
    }
}

public class Circle extends Shape{
    Circle(int l){
        super(l,0);
    }

    @Override
    public void printarea(){
        System.out.println("Area of circle is "+ 3.14*p*p);
    }
}

```

## OUTPUT:

```
C:\Users\trish\OneDrive\Desktop\test>javac Shape.java
C:\Users\trish\OneDrive\Desktop\test>java Shape
Name: Bramha Anilkumar Bajannavar
USN: 1BM23CS071
Area of rectangle is 25
Area of triangle is 5.0
Area of circle is 78.5
C:\Users\trish\OneDrive\Desktop\test>
```

## Program 5:

Develop a Java program to create a class Bank that maintains two kinds of account for its customers, one called savings account and the other current account. The savings account provides compound interest and withdrawal facilities but no cheque book facility. The current account provides cheque book facility but no interest. Current account holders should also maintain a minimum balance and if the balance falls below this level, a service charge is imposed. Create a class Account that stores customer name, account number and type of account. From this derive the classes Cur-acct and Sav-acct to make them more specific to their requirements. Include the necessary methods in order to achieve the following tasks: a) Accept deposit from customer and update the balance. b) Display the balance. c) Compute and deposit interest d) Permit withdrawal and update the balance Check for the minimum balance, impose penalty if necessary and update the balance.

Algorithm:

## LAB - 6

2a/10/24

```
public class Account {
    String customer_name;
    int account_number;
    String account_type;
    int balance;
    Account (String name, int no, String type, int balance) {
        this.customer_name = name;
        this.account_number = no;
        this.account_type = type;
        this.balance = balance;
    }
}
```

g

```
public class curr_act extends Account {
    int min_balance = 5000;
    void deposit();
    curr_act (String name, int no, String type, balance) {
        super (name, no, type, balance);
    }
    void deposit () {
        System.out.println ("Enter deposit amt");
        int a = cin;
        Scanner cin = new Scanner (System.in);
        int a = cin.nextInt();
        balance += a;
    }
    void curr_balance () {
        System.out.println ("Current balance" + balance);
    }
    void check () {
        if (balance < min_balance) {
            balance -= charge;
            System.out.println ("Service charge deducted");
        }
    }
}
```

```

public class Sav-Acc extends Account {
    int balance;
    int min_balance = 5000;
    int charge = 50;
    int interest = 50;
    Sav-Acc (int balance, int account_no, String customer_name,
              String customer_name);
    Sav-Acc (String name, int no, String type, int balance);
    super (name, no, type, balance);
}

void withdraw () {
    System.out.println ("Enter amount");
    Scanner cin = new Scanner (System.in);
    int a = cin.nextInt();
    if (balance > min_balance) {
        System.out.println (a + " rupees was withdrawn");
        balance -= a;
    } else {
        System.out.println ("Not enough balance");
        balance -= charge;
    }
}

void current_balance () {
    System.out.println ("Current balance is " + balance);
}

void add_interest () {
    System.out.println ("After adding interest, balance:");
    balance = balance + balance * interest;
    System.out.println (balance);
}

```

```
public class Bank {  
    public static void main (String [] args) {  
        Account acc1 = new Account ("Branha", 123, "Current", 9000);  
        Account curr_acct = new Current ("Branha", 123, "Current", 9000);  
        Account Sacc1 = new Sav_acct ("Branha", 199, "Savings", 15000);
```

```
        acc1.deposit ();  
        acc1.check ();  
        acc1.cur_balance ();  
        acc2.withdraw ();  
        acc2.addInterest ();  
        acc2.checkBalance ();  
        acc2.withdraw ();  
        acc2.withdraw ();
```

(String trans. acc = 10 - trans)

trans if (General minor < withdraw) {

### Output

Created account for Branha

Enter amount to be deposited

8484

8484 rupees were deposited in your account

balance is 16484.0

Savings account created for Branha at 10.10.2014  
Enter amount to be withdrawn

4545

4545 rupees were ~~debited~~ debited

After adding interest

balance is 20182.5

executed  
29/10/14

Code:

```
import java.util.Scanner;

class Account {
    String customer_name;
    int account_number;
    String account_type;
    int balance;

    Account(String name, int no, String type, int balance) {
        this.customer_name = name;
        this.account_number = no;
        this.account_type = type;
        this.balance = balance;
    }
}

class Curr_act extends Account {
    int min_balance = 5000;
    int charge = 100;

    Curr_act(String name, int no, String type, int balance) {
        super(name, no, type, balance);
    }

    void deposit() {
        System.out.println("Enter deposit amount:");
        Scanner cin = new Scanner(System.in);
        int x = cin.nextInt();
        balance += x;
        System.out.println("Deposit successful!");
    }

    void curr_balance() {
        System.out.println("Current balance: " + balance);
    }

    void check() {
        if (balance < min_balance) {
            balance -= charge;
            System.out.println("Service charge of 100 deducted due to low balance.");
        }
    }
}
```

```

        }
    }
}

class Sav_act extends Account {
    int min_balance = 5000;
    double interest_rate = 5.0;

    Sav_act(String name, int no, String type, int balance) {
        super(name, no, type, balance);
    }

    void withdraw() {
        System.out.println("Enter withdrawal amount:");
        Scanner cin = new Scanner(System.in);
        int x = cin.nextInt();
        if (balance - x < min_balance) {
            System.out.println("Insufficient balance! Minimum balance of 5000 required.");
        } else {
            balance -= x;
            System.out.println("Withdrawal successful!");
        }
    }

    void curr_balance() {
        System.out.println("Current balance: " + balance);
    }

    void add_interest() {
        System.out.println("Adding interest to the account...");
        balance += (int) (balance * (interest_rate / 100));
        System.out.println("Interest added. Current balance: " + balance);
    }
}

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

        Curr_act acc1 = new Curr_act("John", 1234, "Current", 10000);
        Sav_act acc2 = new Sav_act("Doe", 5678, "Savings", 20000);
    }
}

```

```

System.out.println("\n-- Current Account Operations --");
acc1.deposit();
acc1.curr_balance();
acc1.check();
acc1.curr_balance();

System.out.println("\n-- Savings Account Operations --");
acc2.withdraw();
acc2.curr_balance();
acc2.add_interest();
acc2.curr_balance();

}

}

```

OUTPUT:

```

C:\Users\trish\OneDrive\Desktop\test>javac Bank.java

C:\Users\trish\OneDrive\Desktop\test>java Bank
Name: Bramha Anilkumar Bajannavar
USN: 1BM23CS071

-- Current Account Operations --
Enter deposit amount:
30000
Deposit successful!
Current balance: 40000
Current balance: 40000

-- Savings Account Operations --
Enter withdrawal amount:
5000
Withdrawal successful!
Current balance: 15000
Adding interest to the account...
Interest added. Current balance: 15750
Current balance: 15750

C:\Users\trish\OneDrive\Desktop\test>

```

### **Program 6:**

Create a package CIE which has two classes - Personal and Internals. The class Personal has members like usn, name, sem. The class Internals has an array that stores the internal marks scored in five courses of the current semester of the student. Create another package SEE which has the class External which is a derived class of Personal. This class has an array that stores the SEE marks

scored in five courses of the current semester of the student. Import the two packages in a file that declares the final marks of n students in all five courses

Algorithm:

LAB-5  
Date: 12/11/24

```
package CIE;
public class Student {
    public String VSN;
    public String name;
    public int sem;
    public Student (String VSN, String name, int sem) {
        this.VSN = VSN;
        this.name = name;
        this.sem = sem;
    }
    public void display () {
        System.out.println ("Name: " + name);
        System.out.println ("VSN: " + VSN);
        System.out.println ("Sem: " + sem);
    }
}
package CIE;
import java.util.Scanner;
public class Internals extends Student {
    public int [] marks = new int [5];
    public Internals (String VSN, String name, int sem) {
        super (VSN, name, sem);
    }
}
```

```
public void getmarks() {
    Scanner s = new Scanner (System.in);
    for (int i=0; i<5; ++i) {
        System.out.println ("marks");
        marks [i] = s.nextInt();
    }
}
```

```
public void display() {
    System.out.println ("Int - marks");
    for (int i=0; i<5; ++i) {
        System.out.println (marks);
    }
}
```

---

```
package SEE;
import CSE.*;
import java.util.Scanner;
```

```
public class External extends Internals {
    public int [] ent = new int [5];
    public External (String vname, String name, int sem) {
        super (vname, name, sem);
    }
}
```

```
public void getmarks() {
    Scanner s = new Scanner (System.in);
    for (int i=0; i<n; ++i) {
        System.out.println ("Marks");
        marks[i] = s.nextInt();
    }
}
```

```
public void display() {
    System.out.println ("External marks");
    for (int i=0; i<5; ++i) {
        System.out.println (" Marks obtained");
    }
}
```

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

```
public class Main {
    public static void main (String [] args) {
        Scanner S = new Scanner ();
        for (int i=0; i<n; ++i) {
            System.out.println ("Enter details");
            System.out.println ("Name");
            String name = S.nextLine ();
            System.out.println (" USN");
            String USN = S.nextLine ();
            System.out.println ("Sem");
            int sem = S.nextInt ();
        }
    }
}
```

Code:

```
package CIE;

import java.util.Scanner;

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

    public Internals(String usn, String name, int sem) {
        super(usn, name, sem);
    }

    public void getmarks() {
        Scanner s = new Scanner(System.in);
        for (int i = 0; i < 5; i++) {
            System.out.println("Enter internal marks for course " + (i + 1) + ":");
            marks[i] = s.nextInt();
            s.nextLine();
        }
    }

    public void displaymarks() {
        System.out.println("Internal Marks:");
        for (int i = 0; i < 5; i++) {
            System.out.println("Course " + (i + 1) + ": " + marks[i]);
        }
    }
}
```

```
package CIE;
```

```
public class Student {

    public String usn;
    public String name;
    public int sem;

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

```

        this.sem = sem;
    }

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

package SEE;

import CIE.Student;
import CIE.Internals;
import java.util.Scanner;

public class Externals extends Internals {
    public int[] extmarks = new int[5];

    public Externals(String usn, String name, int sem) {
        super(usn, name, sem);
    }

    public void getextmarks() {
        Scanner s = new Scanner(System.in);
        for (int i = 0; i < 5; i++) {
            System.out.println("Enter external marks for course " + (i + 1) + ":");
            extmarks[i] = s.nextInt();
            s.nextLine();
        }
    }

    public void displayextmarks() {
        System.out.println("External Marks:");
        for (int i = 0; i < 5; i++) {
            System.out.println("Course " + (i + 1) + ": " + extmarks[i]);
        }
    }
}

```

```

public void finalmarks() {
    System.out.println("Final Marks (Internal + External):");
    for (int i = 0; i < 5; i++) {
        int finalMark = marks[i] + extmarks[i];
        System.out.println("Course " + (i + 1) + ": " + finalMark);
    }
}
}

```

```

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

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

        for (int i = 0; i < n; i++) {
            System.out.println("\nEnter details for student " + (i + 1));
            System.out.print("Enter USN: ");
            String usn = s.nextLine();
            System.out.print("Enter Name: ");
            String name = s.nextLine();
            System.out.print("Enter Semester: ");
            int sem = s.nextInt();
            s.nextLine();

            Externals student = new Externals(usn, name, sem);

            student.displayStudentInfo();

            System.out.println("Enter internal marks:");
            student.getmarks();
            student.displaymarks();
        }
    }
}

```

```

        System.out.println("Enter external marks:");
        student.getextmarks();
        student.displayextmarks();

        student.finalmarks();
    }

    s.close();
}
}

```

**OUTPUT:**

```

C:\Users\trish\OneDrive\Desktop\test>java MainMarks
Name: Bramha Anilkumar Bajannavar
USN: 1BM23CS071
Enter number of students:
1

Enter details for student 1
Enter USN: 1BM23CS071
Enter Name: Bramha
Enter Semester: 3
USN: 1BM23CS071
Name: Bramha
Semester: 3
Enter internal marks:
Enter internal marks for course 1:
9
Enter internal marks for course 2:
9
Enter internal marks for course 3:
9
Enter internal marks for course 4:
9
Enter internal marks for course 5:
9
Internal Marks:
Course 1: 9
Course 2: 9
Course 3: 9
Course 4: 9
Course 5: 9
Enter external marks:
Enter external marks for course 1:
9
Enter external marks for course 2:
9
Enter external marks for course 3:
9
Enter external marks for course 4:
9
Enter external marks for course 5:
9
External Marks:
Course 1: 9
Course 2: 9
Course 3: 9
Course 4: 9
Course 5: 9
Final Marks (Internal + External):
Course 1: 18
Course 2: 18
Course 3: 18
Course 4: 18
Course 5: 18

C:\Users\trish\OneDrive\Desktop\test>

```

### Program 7:

Create an interface Polygon. It has a default method getPerimeter() and abstract method getArea(). Implement the interface using different shapes.

Algorithm:

→ Polygon Interface

```
interface Polygon {
    default double getPerimeter() {
        return 0;
    }
    abstract double getArea();
}
```

class Rectangle implements Polygon

```
double length;
double width;
public Rectangle(double length, double width) {
    this.length = length;
    this.width = width;
}
@Override
public double getPerimeter() {
    return 2 * (length + width);
}
@Override
public double getArea() {
    return length * width;
}
```

class circle implements Polygon {  
 double radius;  
 public circle (double radius) {  
 this.radius = radius;  
 }  
 *1. function for calculating*

@override

public double getperimeter () {  
 return 2 \* 3.14 \* radius;

}

@override

public double getarea () {  
 return 3.14 \* radius \* radius;

}

public class Main () {

public static void main (String [] args) {

Rectangle r = new Rectangle (4);

System.out.println ("Perimeter " + r.getperimeter());

System.out.println ("Area " + r.getarea());

Circle c = new Circle (4);

System.out.println ("Area of Circle " + c.getperimeter());

System.out.println ("Area of Circle " + c.getarea());

}

*Executed*

O/P

Rectangle perimeter : 16.0

Rectangle area : 15.0

Circle perimeter : 12.566370614359114

Circle area : 50.26548245

Code:

```
interface Polygon {  
  
    default double getPerimeter() {  
        return 0.0;  
    }  
  
    double getArea();  
}  
  
class Rectangle implements Polygon {  
    private double length;  
    private double width;  
  
    public Rectangle(double length, double width) {  
        this.length = length;  
        this.width = width;  
    }  
  
    @Override  
    public double getPerimeter() {  
        return 2 * (length + width);  
    }  
  
    @Override  
    public double getArea() {  
        return length * width;  
    }  
}  
  
class Circle implements Polygon {  
    private double radius;  
  
    public Circle(double radius) {  
        this.radius = radius;  
    }  
  
    @Override  
    public double getPerimeter() {  
        return 2 * Math.PI * radius;  
    }  
}
```

```

    }

    @Override
    public double getArea() {
        return Math.PI * radius * radius;
    }
}

public class Main {
    public static void main(String[] args) {
        Rectangle rectangle = new Rectangle(5, 3);
        System.out.println("Rectangle Perimeter: " + rectangle.getPerimeter());
        System.out.println("Rectangle Area: " + rectangle.getArea());

        Circle circle = new Circle(4);
        System.out.println("Circle Perimeter: " + circle.getPerimeter());
        System.out.println("Circle Area: " + circle.getArea());
    }
}

```

OUTPUT:

```

C:\Users\trish\OneDrive\Desktop\test>javac Main.java
C:\Users\trish\OneDrive\Desktop\test>java Main
Name: Brahma Anilkumar Bajannavar
USN: 1BM23CS071
Rectangle Perimeter: 16.0
Rectangle Area: 15.0
Circle Perimeter: 25.132741228718345
Circle Area: 50.26548245743669

C:\Users\trish\OneDrive\Desktop\test>

```

### Program 8:

Write a program that demonstrates handling of exceptions in inheritance tree. Create a base class called "Father" and derived class called "Son" which extends the base class. In Father class, implement a constructor which takes the age and throws the exception WrongAge( ) when the input age>=father's age

Algorithm:

→ Lab program

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

class Father {  
 int age;  
 public Father (int age) throws WrongAgeException {  
 if (age < 0) {  
 throw new WrongAgeException ("Age cannot be -ve");  
 }  
 this.age = age; }  
 public int getage () { return age; } }  
class Son extends Father {  
 int age;  
 public Son (int fatherage, int Sonage) {  
 super (fatherage);  
 if (Sonage >= fatherage) {  
 throw new WrongException ("Son cannot be older");  
 }  
 this.age = Sonage; } }

```
public class InheritanceExceptionDemo  
public static void main (String [] args) {  
    try {  
        Father f1 = new Father (50);  
        Son s1 = new Son (50, 25);  
        System.out.println ("Test case passed");  
    } catch (WrongAgeException e) {  
        System.out.println ("Error: " + e.getMessage());  
    }  
}
```

```
Father f1 = new Father (50);  
Son s1 = new Son (50, 25);  
System.out.println ("Test case passed");  
Father f2 = new Father (-10);
```

```
catch (WrongAgeException e) {  
    System.out.println ("Error: " + e.getMessage());  
}
```

```
try {  
    Father f3 = new Father (20);  
    Son s2 = new Son (40, 45);  
}
```

```
catch (WrongAgeException e) {  
    System.out.println ("Error: " + e.getMessage());  
}
```

### Output

Test case passed: Father's age = 50

Error: Father's age cannot be negative.

Error: Son's age cannot be greater than the father's

Seen  
26/4/24

Code:

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

class Father {
    int age;
    Father(int age) throws WrongAgeException{
        if(age<0){
            throw new WrongAgeException("Age cannot be negative");
        }
        this.age = age;
    }

    public int getAge() {
        return age;
    }
}

class Son extends Father {
    Son(int fatherAge, int sonAge) throws WrongAgeException {
        super(fatherAge);
        if (sonAge < 0) {
            throw new WrongAgeException("Son's age cannot be negative");
        }
        if (sonAge >= fatherAge) {
            throw new WrongAgeException("Son's age cannot be greater than or equal to father's age");
        }
        this.age = sonAge;
    }
}

public class InheritanceException {
    public static void main(String[] args) {
        try {
            Father f1 = new Father(50);
            Son s1 = new Son(50, 20);
            System.out.println("Test case passed");

            Father f2 = new Father(-1);
        }
    }
}
```

```

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

try {
    Father f3 = new Father(50);
    Son s2 = new Son(50, 60);
} catch (WrongAgeException e) {
    System.out.println(e.getMessage());
}
}
}

```

OUTPUT:

```

C:\Users\trish\OneDrive\Desktop\test>javac InheritanceException.java

C:\Users\trish\OneDrive\Desktop\test>java InheritanceException
Name: Bramha Anilkumar Bajannavar
USN: 1BM23CS071
Test case passed
Age cannot be negative
Son's age cannot be greater than or equal to father's age

C:\Users\trish\OneDrive\Desktop\test>

```

### **Program 9:**

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

Algorithm:

```

Threads know their state, idling
import java.util.concurrent.TimeUnit;
public class ThreadExample {
    public static void main (String args) {
        Thread thread1 = new Thread () {
            while (true) {
                System.out.println ("BMSCE");
                try {
                    TimeUnit.SECONDS.sleep (10);
                } catch (InterruptedException e) {
                    e.printStackTrace();
                }
            }
        };
        thread1.start ();
        Thread thread2 = new Thread () {
            while (true) {
                System.out.println ("CSE");
                try {
                    TimeUnit.SECONDS.sleep (10);
                } catch (InterruptedException e) {
                    e.printStackTrace();
                }
            }
        };
        thread2.start ();
    }
}

```

Output

Bmsce

CSE

CSE

CSE

CSE

CSE

BMSCE

;

;

↳ import javax.swing

public class Integr

    in static vo

Code:

```
public class ThreadExample {  
    public static void main(String[] args) {  
        Thread thread1 = new Thread(()->{  
            while (true) {  
                System.out.println("BMSCE");  
                try {  
                    Thread.sleep(10000);  
                } catch (InterruptedException e) {  
                    e.printStackTrace();  
                }  
            }  
        });  
  
        thread1.start();  
  
        Thread thread2 = new Thread(()->{
```

```
        while (true) {
            System.out.println("CSE");
            try {
                Thread.sleep(2000);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        });
    thread2.start();
}
}
```

OUTPUT:

```
C:\Users\trish\OneDrive\Desktop\test>javac ThreadExample.java
C:\Users\trish\OneDrive\Desktop\test>java ThreadExample
Name: Bramha Anilkumar Bajannavar
USN: 1BM23CS071
BMSCE
CSE
CSE
CSE
CSE
CSE
BMSCE
CSE
CSE
CSE
CSE
CSE
CSE
BMSCE
CSE
CSE
CSE
CSE
CSE
CSE
BMSCE
CSE
CSE
^C
C:\Users\trish\OneDrive\Desktop\test>
```

### Program 10:

Write a program that creates a user interface to perform integer divisions. The user enters two numbers in the text fields, Num1 and Num2. The division of Num1 and Num2 is displayed in the Result field when the Divide button is clicked. If Num1 or Num2 were not an integer, the program would throw a NumberFormatException. If Num2 were Zero, the program would throw an ArithmeticException. Display the exception in a message dialog box.

Algorithm:

Program that creates UI to perform integer division.  
User Interface enters 2 numbers in text fields.  
If either of them is not an integer, raise NumberFormatException at exception.

```
import java.awt.*;  
import java.awt.event.*;  
  
public class Divide {  
    public static void main(String[] args) {  
        JFrame frame = new JFrame();  
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
        frame.setSize(400, 250);  
  
        JPanel panel = new JPanel();  
        panel.setLayout(new GridLayout(4, 1));  
  
        JLabel labelNum1 = new JLabel("Enter Num1");  
        JTextField textFieldNum1 = new JTextField();  
        JLabel labelNum2 = new JLabel("Enter Num2");  
        JTextField textFieldNum2 = new JTextField();  
  
        JButton button = new JButton("Divide");  
  
        public void actionPerformed(ActionEvent e) {  
            try {  
                int num1 = Integer.parseInt(textFieldNum1.getText());  
                int num2 = Integer.parseInt(textFieldNum2.getText());  
            } catch (NumberFormatException e1) {  
                JOptionPane.showMessageDialog(frame, "Please enter integers");  
            } catch (ArithmeticException e2) {  
                JOptionPane.showMessageDialog(frame, "Division by zero is not allowed");  
            }  
        }  
    }  
}
```

if (num2 == 0) {  
 throw new ArithmeticException("Can't divide by 0");  
  
int result = num1 / num2;  
JOptionPane.showMessageDialog(frame, "Result: " + result,  
 "Division Result", JOptionPane.INFORMATION\_MESSAGE);

Catch (NumberFormatException e) {  
 JOptionPane.showMessageDialog(null, "Result : " + result,  
 "Input Error");  
 "Please Enter valid integers");  
}

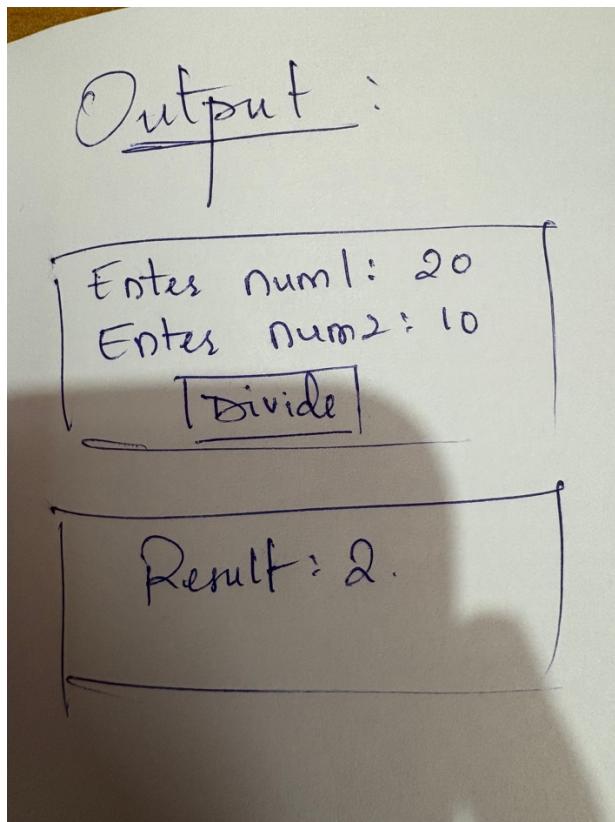
```
catch (ArithmeticException e) {
    JOptionPane.showMessageDialog(frame, "Math Error", "ERROR - MESSAGE");
}
```

```
(y);  
panel.add(labelNum1);  
panel.add(labelNum2);  
panel.add(textNum1);  
panel.add(textNum2);  
panel.add(buttonDivide);
```

~~frame.add(panel);~~ ~~frame.add(true);~~

frame.add(true);

~~Green~~ ~~7/24~~  
~~16/12/23~~



Code:

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

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

        JFrame frame = new JFrame("Integer Division");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setSize(400, 200);
        frame.setLayout(new GridLayout(4, 2));

        JLabel labelNum1 = new JLabel("Enter Num1:");
        JTextField textNum1 = new JTextField();

        JLabel labelNum2 = new JLabel("Enter Num2:");
        JTextField textNum2 = new JTextField();
```

```

JButton divideButton = new JButton("Divide");
JLabel resultLabel = new JLabel("Result: ");

frame.add(labelNum1);
frame.add(textNum1);
frame.add(labelNum2);
frame.add(textNum2);
frame.add(divideButton);
frame.add(resultLabel);

divideButton.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        try {

            int num1 = Integer.parseInt(textNum1.getText());
            int num2 = Integer.parseInt(textNum2.getText());

            int result = num1 / num2;

            resultLabel.setText("Result: " + result);

        } catch (NumberFormatException ex) {

            JOptionPane.showMessageDialog(frame, "Invalid input! Please enter integers.",
                    "NumberFormatException", JOptionPane.ERROR_MESSAGE);
        } catch (ArithmaticException ex) {

            JOptionPane.showMessageDialog(frame, "Cannot divide by zero!",
                    "ArithmaticException", JOptionPane.ERROR_MESSAGE);
        }
    }
});

frame.setVisible(true);
}

```

}

OUTPUT:

```
C:\Users\trish\OneDrive\Desktop\test>javac DivisionApp.java
```

```
C:\Users\trish\OneDrive\Desktop\test>java DivisionApp
```

Name: Bramha Anilkumar Bajannavar

USN: 1BM23CS071

