

VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“JnanaSangama”, Belgaum -590014, Karnataka.



LAB REPORT on

Object Oriented Java Programming (23CS3PCOOJ)

Submitted by

Dama Yohitesh Naveen Sai (**1BM23CS085**)

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 **Dama Yohitesh Naveen Sai (1BM23CS085)**, 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	Dr. Jyothi S Nayak Professor & HOD Department of CSE, BMSCE
--	---

Index

Sl. No.	Date	Experiment Title	Page No.
1	1/10/2024	Lab Program - 1	4-6
2	8/10/2024	Lab Program - 2	7-14
3	15/10/2024	Lab Program - 3	15-20
4	22/10/2024	Lab Program - 4	21-24
5	29/10/2024	Lab Program - 5	25-34
6	12/11/2024	Lab Program – 6	35-42
7	19/11/2024	Lab Program – 7	43-47
8	26/11/2024	Lab Program – 8	48-51
9	3/12/2024	Lab Program – 9	52-55
10	3/12/2024	Lab Program – 10	56-61

Github-link(<https://github.com/Yohitesh/JAVA-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 solutions.

```
import java.util.Scanner  
public class QF {  
    public static void main (String [] args) {  
        Scanner scan = new Scanner (System.in);  
        System.out.println ("Enter the 1st number: ");  
        float a = scan.next (Float());  
        System.out.println ("Enter the 2nd number: ");  
        float b = scan.next (Float());  
        System.out.println ("Enter the 3rd number: ");  
        float c = scan.next (Float());  
        float d = (b*b)-(4*a*c);  
        if (a == 0 & b == 0 & c == 0)  
            System.out.println ("Root Doesn't exist");  
        else {  
            if (d > 0) {  
                float r1 = (-b + Math.sqrt(d))/(2*a);  
                float r2 = (-b - Math.sqrt(d))/(2*a);  
                System.out.println ("Roots are real and distinct  
                    + root1 = " + r1 + " + root2 = " + r2);  
            } else if (d == 0) {  
                float r0 = (-b)/(2*a);  
                System.out.println ("The roots are real and equal  
                    + root1 = " + r0 + " + root2 = " + r0);  
            }  
        }  
    }  
}
```

```

else {
    float real = (-b)/(2*a);
    float img = (Math.sqrt(d))/(2*a);
    System.out.println("The Roots are Imaginary");
    System.out.println("The real root is : " + real +
        " + " + "The imaginary root is : " + img);
}
scan.close();

```

Output:-

Enter the first number : 1.0
 Enter the 2nd number : 2
 Enter the 3rd number : 3

The roots are Imaginary

Root1 : -1.0 + 1.4142135623730951 i
 Root2 : -1.0 - 1.4142135623730951 i

CODE:

```

import java.util.Scanner;
public class QE{
    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        System.out.println("Enter the first number:");
        double a = scan.nextFloat();
        System.out.println("Enter the 2nd number:");
        double b = scan.nextFloat();
        System.out.println("Enter the 3rd number:");
        double c = scan.nextFloat();
        double d = (b * b) - (4 * a * c);
        if (a == 0) {
            System.out.println("Root doesn't exist");
        }
    }
}

```

```

else{
if (d>0) {
double r1=(-b + Math.sqrt(d))/(2*a);
double r2=(-b-Math.sqrt(d))/(2*a);
System.out.println("Roots are real and distinct");
System.out.println("Root 1 = "+r1+" and Root 2 =" +r2);
} else if (d==0) {
double r0=(-b)/(2*a);
System.out.println("The roots are real and equal");
System.out.println("Root 1 Root 2 = "+r0);
} else {
    double real = (-b)/(2*a);
    double img (Math.sqrt(-d))/(2*a); System.out.println("The Roots are Imaginary");
    System.out.println("The real root is "+real+" and the imaginary root is " + img);
    System.out.println("1BM23CS085, Dama Yohitesh Naveen Sai");
}
}
scan.close();
}
}

```

OUTPUT:

```

C:\Users\yohit\OneDrive\Desktop\Java LPs>javac QE.java
C:\Users\yohit\OneDrive\Desktop\Java LPs>java QE.java
Enter the first number:
3
Enter the 2nd number:
2
Enter the 3rd number:
5
The Roots are Imaginary
The real root is -0.3333333333333333 and the imaginary root is 1.247219128924647
1BM23CS085, Dama Yohitesh Naveen Sai

```

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.

```
import java.util.Scanner;
public class Student {
    public static void main (String [] args) {
        char USN[] = "USN";
        char name[] = "Name";
        int size;
        int credits[] = new int [size];
        int marks[] = new int [size];
        credits = new int [size];
        marks = new int [size];
    }
    public void Details () {
        Scanner scan = new Scanner (System.in);
        System.out.println ("Enter USN:");
        USN = scan.nextLine ();
        System.out.println ("Enter name:");
        name = scan.nextLine ();
        System.out.println ("Enter the no. of subjects");
        size = scan.nextInt ();
        credits = new int [size];
        marks = new int [size];
    }
}
```

```

for (int i=0; i<size; i++) {
    System.out.println ("Enter credits of sub " + (i+1));
    credits[i] = scan.nextInt();
    System.out.println ("Enter marks of sub " + (i+1));
    marks[i] = scan.nextInt();
}

public void display() {
    System.out.print("USN: " + usn);
    System.out.print("Name: " + name);
    System.out.print("In Credits & marks details");
    for (int i=0; i<credits.length; i++) {
        System.out.print("In Subject " + (i+1) + " Credits: "
        credits[i] + " Marks: " + marks[i]);
    }
}

```

```

void
public float SCGPA() {
    int t_gp = 0;
    float o_gp = 0;
    System.out.println("Enter total marks grade points : ");
    int credits[] = scan.nextInt();
    for (int i = 0; i < credits.length; i++) {
        switch (marks[i]) {
            case ('A'): t_gp += 10;
            case ('B'): t_gp += 9;
            case ('C'): t_gp += 8;
            case ('D'): t_gp += 7;
            case ('E'): t_gp += 6;
            case ('F'): t_gp += 5;
        }
        if (marks[i] <= 100 && marks[i] > 90) {
            o_gp += credits[i] * 10;
        } else if (marks[i] >= 80 && marks[i] < 90) {
            o_gp += credits[i] * 9;
        } else if (marks[i] >= 70 && marks[i] < 80) {
            o_gp += credits[i] * 8;
        } else if (marks[i] >= 60 && marks[i] < 70) {
            o_gp += credits[i] * 7;
        } else if (marks[i] >= 50 && marks[i] < 60) {
            o_gp += credits[i] * 6;
        } else if (marks[i] >= 40 && marks[i] < 50) {
            o_gp += credits[i] * 5;
        } else {
            o_gp += credits[i] * 0;
        }
    }
    float SCGPA = (o_gp / t_gp) * 10;
    System.out.println("SCGPA of student :" + SCGPA);
}

```

given

System.out.println("SCGPA")

float SCGPA = (o_gp / t_gp) * 10;

System.out.println("SCGPA of student :" + SCGPA);

```
public static void main (String [] args) {  
    Scanner scan = new Scanner (System.in);  
    Student stud = new Student ();  
    stud.Details ();  
    System.out.println ("Enter no. of subjects");  
    int subs = scan.nextInt ();  
    Student stud = new Student (subs);  
    stud.Details ();
```

```
stud.display ();  
scan.close ();
```

}

(0 < i < n) = true
(0 < i < n) > p (0 < n) <= q;

Output

Enter no. of Subjects : 3

Enter the details for Student :

Enter USN : 1BM23CS086

Enter name : ~~Ram~~ Yohitesh

Enter Credits for Subject 1 : 4

Enter Marks for Subject 1 : 90

Enter Credits for subject 2 : 3

Enter marks for subject 2 : 85

Enter credits for subject 3 : 3

Enter marks for subject 3 : 92

USN : 1BM23CS086

Student Name : Ram

Subject 1 - Credits : 4 , Marks : 90

Subject 2 - Credits : 3 , Marks : 85

Subject 3 - Credits : 3 , Marks : 92

SPPA : ~~9.65~~ 9.7

CODE:

```
import java.util.Scanner;

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

    public void cred_marks(int size) {
        credits = new int[size];
        marks = new int[size];
    }

    public void details() {
        Scanner scan = new Scanner(System.in);
        System.out.println("Enter usn:");
        usn = scan.nextLine().toCharArray();
        System.out.println("Enter name:");
        name = scan.nextLine();
        System.out.println("Enter the no. of subjects:");
        int size = scan.nextInt();
        credits = new int[size];
        marks = new int[size];
        for (int i = 0; i < size; i++) {
            System.out.println("Enter credits of subject " + (i + 1) + ":");
            credits[i] = scan.nextInt();
            System.out.println("Enter marks of subject " + (i + 1) + ":");
            marks[i] = scan.nextInt();
        }
    }

    public void display() {
        System.out.println("USN: " + String.valueOf(usn));
        System.out.println("Name: " + name);
        System.out.println("Credits and Marks details:");
        for (int i = 0; i < credits.length; i++) {
            System.out.println("Subject " + (i + 1) + " - Credits: " + credits[i] + ", Marks: "
+ marks[i]);
        }
    }
}
```

```

}

public void sgpa() {
    Scanner scan = new Scanner(System.in);
    System.out.println("Enter total no of grade points:");
    int t_gp = scan.nextInt();
    int o_gp = 0;
    for (int i = 0; i < credits.length; i++) {
        if (marks[i] <= 100 && marks[i] >= 90) {
            o_gp += credits[i] * 10;
        } else if (marks[i] >= 80 && marks[i] < 90) {
            o_gp += credits[i] * 9;
        } else if (marks[i] >= 70 && marks[i] < 80) {
            o_gp += credits[i] * 8;
        } else if (marks[i] >= 60 && marks[i] < 70) {
            o_gp += credits[i] * 7;
        } else if (marks[i] >= 50 && marks[i] < 60) {
            o_gp += credits[i] * 6;
        } else if (marks[i] >= 40 && marks[i] < 50) {
            o_gp += credits[i] * 5;
        } else {
            o_gp += credits[i] * 0;
        }
    }
    double sgpa = (double) o_gp / t_gp;
    System.out.println("SGPA of student: " +
        sgpa*10);
}

```

```

public static void main(String[] args) {
    Scanner scan = new Scanner(System.in);
    System.out.println("1BM23CS085, Dama Yohitesh Naveen Sai");
    System.out.println("Enter no of subjects:");
    int subs = scan.nextInt();
    Student stud = new Student();
    stud.cred_marks(subs);
    stud.details();
    stud.display();
    stud.sgpa();
    scan.close();
}

```

OUTPUT:

```
C:\Users\yohit\OneDrive\Desktop\Java LPs>java Student.java
1BM23CS085, Dama Yohitesh Naveen Sai
Enter no of subjects:
3
Enter usn:
1BM23CS085
Enter name:
Yohitesh
Enter the no. of subjects:
3
Enter credits of subject 1:
4
Enter marks of subject 1:
90
Enter credits of subject 2:
3
Enter marks of subject 2:
85
Enter credits of subject 3:
3
Enter marks of subject 3:
92
USN: 1BM23CS085
Name: Yohitesh
Credits and Marks details:
Subject 1 - Credits: 4, Marks: 90
Subject 2 - Credits: 3, Marks: 85
Subject 3 - Credits: 3, Marks: 92
Enter total no of grade points:
100
SGPA of student: 9.7
```

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.

→ develop a java program to create n book objects

```
import java.util.Scanner;
public class Book {
    private String name;
    private String Author;
    private double Price;
    private int num_pages;
    // constructor
    public void Book (String name, String Author, double Price, int num_pages) {
        this.name = name;
        this.Author = Author;
        this.Price = Price;
        this.num_pages = num_pages;
    }
    public void get_details () {
        System.out.println ("Name:" + name + "Author:" + Author +
                           "Price:" + Price + "In num-pages:" + num_pages);
    }
    public void set_details (String name, String Author, double Price,
                           int num_pages) {
        this.name = name;
        this.Author = Author;
        this.Price = Price;
        this.num_pages = num_pages;
    }
}
```

```
// overriding - overrides the constructor.
public String toString() {
    return "Name: " + name + " | Author: " + author + " | price: " + price +
           " | Num-pages: " + num_pages;
}
```

```
public static void main(String[] args) {
    Scanner scan = new Scanner(System.in);
    System.out.println("Enter no. of books: ");
    int n = scan.nextInt();
    Book[] books = new Book[n];
    for (int i=0; i<n; i++) {
        System.out.println("Enter details of book: " + (i+1));
        System.out.print("Enter name of the book: ");
        String name = scan.nextLine();
        System.out.print("Enter Author of the book: ");
        String author = scan.nextLine();
        System.out.print("Enter price of the book: ");
        double price = scan.nextDouble();
        System.out.print("Enter no. of pages of the book: ");
        int num_pages = scan.nextInt();
    }
}
```

```
books[i] = new Book(name, author, price, num_pages);
}
System.out.println("Details of all books: ");
for (int i=0; i<n; i++) {
    System.out.println("Details of Book " + (i+1) + ":");
    System.out.println(books[i].toString());
}
}
scan.close();
```

*see
execute*

OUTPUT:

15

Output :-

Enter the no. of books : 2

Enter details of book 1 :

Enter name of the book : Honor

Enter Author of the book : Dhruva

Enter price of the book : 100

Enter no. of pages of the book : 0

Enter details of book 2 :

Enter name of the book : Hello

Enter Author of the book : Walker

Enter price of the book : 255

Enter no. of pages of the book : 500

Details of all Books :

Details of book 1 :

Name : Honor

Author : Dhruva

Price : 100

Num-pages : 0

Details of book 2 :

Name : Hello

Author : Walker

Price : 255

Num-pages : 500

Ques Seen
88
15/10/24

CODE:

```
import java.util.Scanner;

class Book {
    private String name;
    private String author;
    private double price;
    private int num_pages;

    public Book(String name, String author, double price, int num_pages) {
        this.name = name;
        this.author = author;
        this.price = price;
        this.num_pages = num_pages;
    }
    public void get_details(){
        System.out.println("Book Name: " + name + ", Author: " + author + ", Price: " +
price + ", Pages: " + num_pages);
    }
    public void set_details(String name, String author, double price, int num_pages) {
        this.name = name;
        this.author = author;
        this.price = price;
        this.num_pages = num_pages;
    }

    @Override
    public String toString() {
        return "Book Name: " + name + ", Author: " + author + ", Price: " + price + ",
Pages: " + num_pages;
    }
}

public class BookTest {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.println("1BM23CS085, Dama Yohitesh Naveen Sai");
        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 details for book " + (i + 1) + ":");

    System.out.print("Name: ");
    String name = scanner.nextLine();

    System.out.print("Author: ");
    String author = scanner.nextLine();

    System.out.print("Price: ");
    double price = scanner.nextDouble();

    System.out.print("Number of pages: ");
    int num_pages = scanner.nextInt();
    scanner.nextLine(); // Consume newline

    books[i] = new Book(name, author, price, num_pages);
}

System.out.println("\nDetails of the books:");
for (Book book : books) {
    System.out.println(book);
}

scanner.close();
}
```

OUTPUT:

```
C:\Users\yohit\OneDrive\Desktop\Java LPs>javac BookTest.java
C:\Users\yohit\OneDrive\Desktop\Java LPs>java BookTest.java
IBM23CS085, Dama Yohitesh Naveen Sai
Enter the number of books:
2
Enter details for book 1:
Name: Honor
Author: Dhruva
Price: 1
Number of pages: 0
Enter details for book 2:
Name: Hello
Author: Walker
Price: 255
Number of pages: 500

Details of the books:
Book Name: Honor, Author: Dhruva, Price: 1.0, Pages: 0
Book Name: Hello, Author: Walker, Price: 255.0, Pages: 500
```

PROGRAM 4

Create an abstract class Animal with two methods, eat and sleep. Create three subclasses Lion, Deer, and Tiger that extend the Animal class and implement the eat and sleep methods based on their specific behavior. Write a Main class to create objects of these subclasses and demonstrate their behavior by calling the eat and sleep methods. Include your name and USN in the output.

```
import java.util.Scanner;
public abstract class Animal {
    public abstract void eat();
    public abstract void sleep();
}

public class Lion extends Animal {
    @Override
    public void eat() {
        System.out.println("Hunt");
    }
    @Override
    public void sleep() {
        System.out.println("Sleeping");
    }
}

public class Deer extends Animal {
    @Override
    public void eat() {
        System.out.println("Grazes");
    }
    @Override
    public void sleep() {
        System.out.println("Deer Resting");
    }
}
```

```

public class Tiger extends Animal {
    @Override
    public void eat() {
        System.out.println("Tiger is eating!");
    }
    @Override
    public void sleep() {
        System.out.println("Tiger sleeping!");
    }
}

```

```

public class Client {
    public static void main(String args[]) {
        Animal t = new Lion();
        t.sleep();
        t.eat();

        Animal t2 = new Tiger();
        t2.sleep();
        t2.eat();

        Animal D = new Deer();
        D.sleep();
        D.eat();
    }
}

```

~~seen execution~~

3
Output:

Lion is eating!	Hunt
Lion is sleeping	Sleeping
Deer is eating!	Cirrus
Deer is sleeping	Deer is Resting
Tiger is eating!	Tiger is eating!
Tiger is sleeping	Tiger sleeping!

OP seen

22/10/24

CODE:

```
abstract class Animal {  
    abstract void eat();  
  
    abstract void sleep();  
}  
  
class Lion extends Animal {  
    @Override  
    void eat() {  
        System.out.println("Hunt");  
    }  
  
    @Override  
    void sleep() {  
        System.out.println("Sleeping");  
    }  
}  
  
class Deer extends Animal {  
    @Override  
    void eat() {  
        System.out.println("Grass");  
    }  
  
    @Override  
    void sleep() {  
        System.out.println("Deer Resting");  
    }  
}  
  
class Tiger extends Animal {  
    @Override  
    void eat() {  
        System.out.println("Tiger is eating!");  
    }  
}
```

```

    }

    @Override
    void sleep() {
        System.out.println("Tiger Sleeping");
    }
}

public class Main4 {
    public static void main(String[] args) {
        System.out.println("1BM23CS085, Dama Yohitesh Naveen Sai");
        Animal l = new Lion();
        l.eat();
        l.sleep();
        Animal d = new Deer();
        d.eat();
        d.sleep();
        Animal t = new Tiger();
        t.eat();
        t.sleep();
    }
}

```

OUTPUT:

```

C:\Users\yohit\OneDrive\Desktop\Java LPs>javac Main4.java

C:\Users\yohit\OneDrive\Desktop\Java LPs>java Main4.java
1BM23CS085, Dama Yohitesh Naveen Sai
Hunt
Sleeping
Grass
Deer Resting
Tiger is eating!
Tiger Sleeping

```

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.

Demonstration how writing
import java.util.Scanner;

```
public class Account {
    String[] customername;
    int[] Accountnumber;
    String[] Accounttype;
    double[] Balance;
    public void Account(String Customername, int Accountnumber,
                        String Accounttype) {
        Customername[0] = this.Customername;
        Accountnumber[0] = this.Accountnumber;
        Accounttype[0] = this.Accounttype;
        Balance[0] = this.Balance;
    }
}
```

```
public void deposit (double amount) {  
    Balance += amount;  
    System.out.println ("Deposit successful! " + Balance);
```

```
}  
public void displayBalance () {  
    System.out.println ("Balance = " + Balance);
```

```
public class SavAccount extends Account {  
    final double interest = 0.05;
```

```
    public void SavAcc (String customerName, int accountNumber,  
        double balance) {  
        Super (customerName, accountNumber, balance);
```

```
}  
public void compoundInterest () {  
    balance += balance * Math.pow (1 + interest, 1);
```

```
public void withdraw (double amount) {
```

```
    if (balance >= amount) {
```

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

```
        balance -= amount;  
        System.out.println ("Withdrawal successful! " +
```

```
        balance);
```

```
}  
else {  
    System.out.println ("Insufficient balance");
```

26

```

public class Current extends Account {
    double min-bal = 500;
    double servicecharge = 50;

    public Current (String CustomerName, int AccountNumber,
                    double Balance) {
        super (CustomerName, AccountNumber, "Current", Balance);
    }

    public void minbalance () {
        if (Balance < min-bal) {
            Balance -= servicecharge;
            System.out.println ("System.out.println ("balance is less than
min-balance so servicecharge of " + servicecharge + " is imposed");
        }
    }

    public void withdraw (double amount) {
        if (Balance >= amount) {
            Balance -= amount;
            if (Balance < min-bal) {
                Balance -= servicecharge;
                System.out.println ("balance is less than
min-balance so servicecharge of " + servicecharge + " is imposed");
            }
        }
    }

    System.out.println ("Withdrawal Successful, Balance=" + Balance);
}

class CurrentTest {
    public static void main (String args[]) {
        Current c = new Current ("John Doe", 1234567890, 1000);
        c.withdraw (500);
        System.out.println ("Insufficient Balance");
    }
}

```

```

public class Bank {
    public static void main (String [] args) {
        Scanner scan = new Scanner (System.in);
        System.out.println ("Enter no. of customers:");
        int n = scan.nextInt();
        for (int i=0; i<n; i++) {
            System.out.println ("1. saving, 2. current");
            int ch = scan.nextInt();
            System.out.println ("Enter customer-name");
            String name = scan.nextLine();
            System.out.println ("Enter Acc.no.");
            int accno = scan.nextInt();
            System.out.println ("Enter Balance");
            double balance = scan.nextDouble();
            switch (ch) {
                case 1:
                    Sav-acct sav = new Sav-acct (customerName);
                    AccountNumber[i], Balance[i]);
                    sav.displayBalance();
                    System.out.println ("Enter deposit amount");
                    sav.deposit (scan.nextDouble());
                    System.out.println ("Enter compound interest");
                    sav.compoundInterest();
                    System.out.println ("Enter withdrawal amount");
                    sav.withdraw (scan.nextDouble());
                    sav.displayBalance();
            }
        }
    }
}

```

Case 2: {

Cur acct cur = new Cur-acct (Customername [i],
Accountnumber [i], Balance [i]);

cur.displaybalance();

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

cur.deposit ~~Scanner~~(Scan, next, Double());

```
System.out.print("Enter withdrawal amount:");
```

cur.withdraw(); Scan.nextDouble();

```
cur.displayBalance();
```

0.7% formaldehyde through 0.1% to fixate.

Default Locationways (located at [http://127.0.0.1:8000/api/v1/locationways/](#))

`System.out.println("Invalid choice");`

exit(1); // exit with error code

96

Output

Enter no. of accounts : 2

Choose account type : 1. Saving, 2. Current

1
Enter customer name : Danish

Enter account number : 1

Enter initial deposit : 1000

Balance deposit amount : 500

Deposit successful. New Balance: 1500.0

Interest of 75.0 deposited. New Balance: 1575.0

Enter withdrawal amount : 200

Withdrawal successful. New balance: 1375.0

Balance for danish : 1375.0.

Choose account type : 1. Saving 2. Current

2

Enter customer name : Naveen

Enter account number : 2

Enter initial deposit : 3000

Balance for Naveen : 3000.0

Enter deposit amount : 1000

OP Seen
Deposit successful. New balance: 4000.0

Enter withdrawal amount : 500

Withdrawal successful. New balance : 3500.0

Balance for Naveen : 3500.0

*YK
12/11/13*

CODE:

```
import java.util.Scanner;
class Account {
    String customerName;
    String accountNumber;
    String accountType;
    double balance;

    public Account(String customerName, String accountNumber, String accountType,
double initialBalance) {
        this.customerName = customerName;
        this.accountNumber = accountNumber;
        this.accountType = accountType;
        this.balance = initialBalance;
    }

    public void deposit(double amount) {
        balance += amount;
        System.out.println("Deposit successful. New balance: " + balance);
    }

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

class Sav_acct extends Account {
    double interestRate = 0.05;
    public Sav_acct(String customerName, String accountNumber, double
initialBalance) {
        super(customerName, accountNumber, "Savings", initialBalance);
    }

    public void computeAndDepositInterest() {
        double interest = balance * interestRate;
        balance += interest;
        System.out.println("Interest of " + interest + " deposited. New balance: " +
balance);
    }
}
```

```

}

public void withdraw(double amount) {
    if (balance >= amount) {
        balance -= amount;
        System.out.println("Withdrawal successful. New balance: " + balance);
    } else {
        System.out.println("Insufficient balance for withdrawal.");
    }
}

class Cur_acct extends Account {
    double minimumBalance = 500;
    double serviceCharge = 50;
    public Cur_acct(String customerName, String accountNumber, double
initialBalance) {
        super(customerName, accountNumber, "Current", initialBalance);
    }

    public void withdraw(double amount) {
        if (balance >= amount) {
            balance -= amount;
            if (balance < minimumBalance) {
                balance -= serviceCharge;
                System.out.println("Balance below minimum. Service charge of " +
serviceCharge + " imposed.");
            }
            System.out.println("Withdrawal successful. New balance: " + balance);
        } else {
            System.out.println("Insufficient balance for withdrawal.");
        }
    }
}

public class Bank {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("1BM23CS085, Dama Yohitesh Naveen Sai\n");
    }
}

```

```

System.out.println("Enter the number of customers: ");
int numCustomers = scanner.nextInt();
scanner.nextLine();

for (int i = 0; i < numCustomers; i++) {
    System.out.println("\nCustomer " + (i + 1) + ":");

    System.out.println("Choose account type: 1 for Savings, 2 for Current");
    int ch = scanner.nextInt();
    scanner.nextLine();

    System.out.print("Enter customer name: ");
    String name = scanner.nextLine();

    System.out.print("Enter account number: ");
    String accNum = scanner.nextLine();

    System.out.print("Enter initial deposit: ");
    double initialDeposit = scanner.nextDouble();

    switch (ch) {
        case 1:
            Sav_acct savings = new Sav_acct(name, accNum, initialDeposit);
            savings.displayBalance();
            System.out.print("Enter deposit amount: ");
            savings.deposit(scanner.nextDouble());
            savings.computeAndDepositInterest();
            System.out.print("Enter withdrawal amount: ");
            savings.withdraw(scanner.nextDouble());
            savings.displayBalance();
            break;
        case 2:
            Cur_acct current = new Cur_acct(name, accNum, initialDeposit);
            current.displayBalance();
            System.out.print("Enter deposit amount: ");
            current.deposit(scanner.nextDouble());
            System.out.print("Enter withdrawal amount: ");
            current.withdraw(scanner.nextDouble());
            current.displayBalance();
            break;
    }
}

```

```

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

    scanner.close();
}
}

```

OUTPUT:

```

C:\Users\yohit\OneDrive\Desktop\Java LPs>javac Bank.java

C:\Users\yohit\OneDrive\Desktop\Java LPs>java Bank.java
1BM23CS085, Dama Yohitesh Naveen Sai
Enter the number of customers:
2

Customer 1:
Choose account type: 1 for Savings, 2 for Current
1
Enter customer name: Danish
Enter account number: 1
Enter initial deposit: 1000
Balance for Danish: 1000.0
Enter deposit amount: 500
Deposit successful. New balance: 1500.0
Interest of 75.0 deposited. New balance: 1575.0
Enter withdrawal amount: 700
Withdrawal successful. New balance: 875.0
Balance for Danish: 875.0

Customer 2:
Choose account type: 1 for Savings, 2 for Current
2
Enter customer name: Naveen
Enter account number: 2
Enter initial deposit: 3000
Balance for Naveen: 3000.0
Enter deposit amount: 500
Deposit successful. New balance: 3500.0
Enter withdrawal amount: 400
Withdrawal successful. New balance: 3100.0
Balance for Naveen: 3100.0

```

PROGRAM 6

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

```
/// File name → Student.java
package CIE;

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

    public Internals (int[] marks) {
        this.marks = marks;
    }
}
```

```

// External.java
package SEE;
import CIE.Student;

public class External extends Student {
    public int[] marks = new int[5];
    public External (String USN, String name, int sem, int[] marks) {
        super (USN, name, sem);
        this.marks = marks;
    }
}

```

// Mainprogram.java

```

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

public class MyCode {
    public static void main (String [] args) {
        Scanner scan = new Scanner (System.in);
        System.out.print ("Enter no. of Students:");
        int n = scan.nextInt ();
        External [] students = new External [n];
        Internal [] internals = new Internal [n];
        for (int i=0; i<n; i++) {
            System.out.println ("In Student " + (i+1));
            System.out.println ("USN:");
            String USN = scan.next();
            System.out.println ("Name:");
            String name = scan.next();
        }
    }
}

```

```

System.out.println("Semester:");
int sem = sc.nextInt();
System.out.print("Internal Marks (5 subs):");
int[] intMarks = new int[5];
for (int j = 0; j < 5; j++) {
    intMarks[j] = sc.nextInt();
}
internal[i] = new Internal(intMarks);
System.out.print("SEE marks (5 subs):");
int extMarks = new int[5];
for (int j = 0; j < 5; j++) {
    extMarks[j] = sc.nextInt();
}
students[i] = new External(USN, name, sem, extMarks);

```

```

System.out.println("Infinal Marks:");
for (int i = 0; i < n; i++) {
    System.out.println(students[i].name + " " +
        students[i].USN + ")");
    for (int j = 0; j < 5; j++) {
        finalMark = internal[i].marks[j] + (students[i].marks[j] / 2);
        System.out.print("Course" + (j + 1) + ":" + finalMark);
    }
}

```

System.out.println();

sc.close()

}

Q1 Output

Enter number of students: 2

Enter USN: 16M23CS085

Enter Name: Yohitesh

Enter Semester: 3

Enter 5 internal marks:

45

44

50

48

48

48

48

Enter SEE Marks:

50

50

48

49

50

Enter USN: 16M23CS086

Enter Name: Danish

Enter Semester: 3

Enter 5 internal marks:

45

46

47

48

50

Enter 5 SEE Marks:

50

50

49

50

49

Final Marks:

Student 1: 45 44 50 48 48 48 48 48 48 48 50

USN: 16M23CS085

Name: Yohitesh

Semester: 3

Final Marks in 5 Courses:

95 96 95 97 100

Student 2: 45 44 50 48 48 48 48 48 48 48 50

USN: 16M23CS086

Name: Danish

Semester: 3

Final Marks in 5 Courses

95 96 96 98 99

CODE:

Personal.java

package CIE;

```
public class Personal
{
    public String USN;
    public     String
    name;  public  int
    sem;
```

```
public Personal(String USN,String name,int sem)
```

```
{
    this.USN = USN;
    this.name = name;
    this.sem = sem;
}
```

```
}
```

Internals.java

package CIE;

```
public class Internals extends Personal {
```

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

```
    public Internals(String usn, String name, int sem, int[] internalMarks) {
```

```
        super(usn, name, sem);
        this.internalMarks = internalMarks;
    }
```

```
}
```

Externals.java

package SEE;

```
import CIE.Personal;
```

```
public class External extends Personal {
```

```
    public int[] seeMarks;
```

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

Main6.java

```
import CIE.Internals;  
import SEE.External;  
import java.util.Scanner;  
  
public class Main6 {  
    public static void main(String[] args) {  
        Scanner sc = new Scanner(System.in);  
        System.out.print("1BM23CS085, Dama Yohitesh Naveen Sai\n");  
  
        System.out.print("Enter the number of students: ");  
        int n = sc.nextInt();  
  
        Internals[] internalStudents = new Internals[n];  
        External[] externalStudents = new External[n];  
  
        for (int i = 0; i < n; i++) {  
            sc.nextLine(); // Clear the buffer  
  
            System.out.print("Enter USN: ");  
            String usn = sc.nextLine();  
            System.out.print("Enter Name: ");  
            String name = sc.nextLine();  
            System.out.print("Enter Semester: ");  
            int sem = sc.nextInt();  
  
            int[] internalMarks = new int[5];  
            System.out.println("Enter 5 internal marks:");  
            for (int j = 0; j < 5; j++) {  
                internalMarks[j] = sc.nextInt();  
            }  
  
            internalStudents[i] = new Internals(usn, name, sem, internalMarks);  
            int[] seeMarks = new int[5];  
        }  
    }  
}
```

```

System.out.println("Enter 5 SEE marks:");
for (int j = 0; j < 5; j++) {
    seeMarks[j] = sc.nextInt();
}

externalStudents[i] = new External(usn, name, sem, seeMarks);
}
System.out.println("\nFinal
Marks:"); for (int i = 0; i < n; i++) {
    System.out.println("Student " + (i + 1) + ":");
    System.out.println("USN: " + internalStudents[i].USN);
    System.out.println("Name: " + internalStudents[i].name);
    System.out.println("Semester: " + internalStudents[i].sem);

    System.out.print("Final Marks in 5 Courses: ");
    for (int j = 0; j < 5; j++) {
        // Calculate final mark for each course
        int finalMark = internalStudents[i].internalMarks[j] +
externalStudents[i].seeMarks[j];
        System.out.print(finalMark + " ");
    }
    System.out.println("\n");
}

sc.close();
}

}

```

OUTPUT 1:

```
C:\Users\yohit\OneDrive\Desktop\Java LPs\Lab Program 6>javac -d . Main6.java  
C:\Users\yohit\OneDrive\Desktop\Java LPs\Lab Program 6>java Main6  
1BM23CS085, Dama Yohitesh Naveen Sai  
Enter the number of students: 2  
Enter USN: 1BM23CS085  
Enter Name: Yohitesh  
Enter Semester: 3  
Enter 5 internal marks:  
45  
44  
50  
48  
48  
Enter 5 SEE marks:  
50  
50  
49  
48  
50  
Enter USN: 1BM23CS086  
Enter Name: Danish  
Enter Semester: 3  
Enter 5 internal marks:  
45  
46  
47  
48  
50  
Enter 5 SEE marks:  
50  
50  
49  
50  
49
```

OUTPUT 2:

```
Final Marks:  
Student 1:  
USN: 1BM23CS085  
Name: Yohitesh  
Semester: 3  
Final Marks in 5 Courses: 95 94 99 96 98  
  
Student 2:  
USN: 1BM23CS086  
Name: Danish  
Semester: 3  
Final Marks in 5 Courses: 95 96 96 98 99
```

PROGRAM 7

Write a Java program to create an interface Polygon with methods getArea() and getPerimeter(). Implement two classes, Rectangle and Circle, to calculate the area and perimeter of a rectangle and a circle. Use a Main class to test these implementations and display the results along with your name and USN.

```
→ import java.util.Scanner  
→ Interface Polygon {  
    default void getPerimeter()  
    default double getPerimeter () {  
        return 0.0;  
    }  
    double getArea ();  
}  
  
public class Rectangle implements Polygon {  
    Rectangle () {  
        double length;  
        double width;  
        Rectangle (double length, double width) {  
            this.length = length;  
            this.width = width;  
        }  
    }  
}
```

@Override

getPerimeter

public double getPerimeter() {

return 2 * (length + width);

}

@Override

public double getArea() {

return length * width;

}

3
public class Circle implements Polygon {

double radius; ;

Circle (double radius) {

this.radius = radius;

}

~~public~~ @Override

public double getPerimeter() {

return 2 * 3.14 * radius;

}

@Override Override

public double getArea() {

return 3.14 * radius * radius;

3

3

```

public static void main()
{
    public class MainLab7 {
        public static void main (String [] args) {
            Scanner scan = new Scanner (System.in);
        }
    }

    Rectangle rect = new
    System.out.println ("Enter length of Rectangle:");
    double len = scan.nextDouble();
    System.out.println ("Enter width of Rectangle:");
    double wid = scan.nextDouble();
    Rectangle rect = new Rectangle (len, wid);
    System.out.println ("Perimeter = " + rect.getPerimeter());
    System.out.println ("Area = " + rect.getArea());
}

System.out.println ("Enter radius of circle:");
double rad = scan.nextDouble();
Circle c = new Circle (rad);
System.out.println ("Perimeter = " + c.getPerimeter());
System.out.println ("Area = " + c.getArea());
}

```

Output:

```

Enter length of rectangle: 10.0
Enter breadth of rectangle: 10.0
Perimeter = 40.0
Area = 100.0
Enter radius of circle: 4.0
Perimeter = 25.1327
Area = 50.2654

```

~~Scen~~
~~25.1327~~
~~50.2654~~

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 getArea() {  
        return length * width;  
    }  
  
    @Override  
    public double getPerimeter() {  
        return 2 * (length + width);  
    }  
}  
  
class Circle implements Polygon {  
    private double radius;  
  
    public Circle(double radius) {  
        this.radius = radius;  
    }  
}
```

```

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

@Override
public double getPerimeter() {
    return 2 * Math.PI * radius;    }
}

public class Main7 {
    public static void main(String[] args) {
        System.out.print("1BM23CS085, Dama Yohitesh Naveen Sai\n");

        Rectangle rectangle = new Rectangle(5, 3);
        System.out.println("Rectangle Area: " + rectangle.getArea());
        System.out.println("Rectangle Perimeter: " + rectangle.getPerimeter());

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

```

OUTPUT:

```

C:\Users\yohit\OneDrive\Desktop\Java LPs\Lab Program 7>javac Main7.java

C:\Users\yohit\OneDrive\Desktop\Java LPs\Lab Program 7>java Main7
1BM23CS085, Dama Yohitesh Naveen Sai
Rectangle Area: 15.0
Rectangle Perimeter: 16.0
Circle Area: 50.26548245743669
Circle Perimeter: 25.132741228718345

```

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<0. In Son class, implement a constructor that uses both father and son's age and throws an exception if son's age is >=father's age.

```
Wrong Age class extends Exception {  
    public WrongAge (String msg) {}  
    super (msg);  
  
class Father {  
    int age;  
    public Father (int age) throws WrongAge {  
        if (age < 0) {  
            throw new WrongAge ("Father age can't be negative");  
        }  
        this.age = age;  
        System.out.println ("Father age is set to: " + this.age);  
  
class Son extends Father {  
    int sonage;  
    public Son (int sonage) throws WrongAge {  
        if (sonage >= age) {  
            throw new WrongAge ("Son's age can't be more than  
            or equal to Father's age");  
        }  
        this.sonage = sonage;  
        System.out.println ("Son's age is set to: " + this.sonage);  
    }
```

```

class Son extends Father {
    int son_age; // son's age can't be less than father's age
    public Son (int son_age, int f_age) throws WrongAge {
        super (f_age);
        if (son_age >= f_age) {
            throw new WrongAge ("Son's age can't be greater than or equal to father's age");
        }
        this.son_age = son_age;
        System.out.println("Son's age is set to: " + this.son_age);
    }
}

public class Main {
    public static void main (String args[]) {
        try {
            Son son1 = new Son (40, 45); // invalid (wrong)
        } catch (WrongAge e) {
            System.out.println("Exception Caught: " + e.getMessage());
        }
        try {
            Son son2 = new Son (20, 40);
        } catch (WrongAge e) {
            System.out.println("Exception Caught: " + e.getMessage());
        }
    }
}

```

Output: (20, 40)

~~seen
solution~~

Father's age is set to 40

Exception caught: Son's age cannot be greater than or equal to father's age

Father's age is set to 40

Son's age is set to 20

CODE:

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

```
public class Main8 {  
    public static void main(String[] args) {  
        System.out.print("1BM23CS085, Dama Yohitesh Naveen Sai\n");  
        try {  
            Son son1 = new Son(40, 45);  
        } catch (WrongAge e) {  
            System.out.println("Exception caught: " + e.getMessage());  
        }  
  
        try {  
            Son son2 = new Son(40, 30);  
        } catch (WrongAge e) {  
            System.out.println("Exception caught: " + e.getMessage());  
        }  
    }  
}
```

OUTPUT:

```
C:\Users\yohit\OneDrive\Desktop\Java LPs>javac Main8.java  
  
C:\Users\yohit\OneDrive\Desktop\Java LPs>java Main8.java  
1BM23CS085, Dama Yohitesh Naveen Sai  
Father's age is set to 40  
Exception caught: Son's age cannot be greater than or equal to father's age.  
Father's age is set to 40  
Son's age is set to 30
```

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.

```
class Displaymsg & extends Thread {  
    private String msg;  
    private int interval;  
  
    public Displaymsg (String msg, int interval) {  
        this.msg = msg;  
        this.interval = interval;  
    }  
    public void Run() {  
        try {  
            while (true) {  
                System.out.println (msg);  
                Thread.sleep (interval * 1000);  
            }  
        } catch (InterruptedException e) {  
            System.out.println ("Thread Interrupted "+ e.getMessage());  
        }  
    }  
}  
  
public class Threaddemo {  
    public static void main (String args[]) {  
        Thread th1 = new Displaymsg ("BMS College of Engineering", 10);  
        Thread th2 = new Displaymsg ("CSE", 2);  
  
        th1.start();  
        th2.start();  
    }  
}
```

Output

BMS College of Engineering

CSE

CSE

CSE

CSE

CSE

BMS College of Engineering

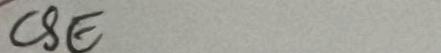
CSE

CSE

CSE

CSE

CSE



CSE

CSE

1C

CODE:

```
class DisplayMessage extends Thread {  
    private String message;  
    private int interval;  
  
    public DisplayMessage(String message, int interval) {  
        this.message = message;  
        this.interval = interval;  
    }  
  
    public void run() {  
        try {  
            while (true) {  
                System.out.println(message);  
                Thread.sleep(interval * 1000);  
            }  
        } catch (InterruptedException e) {  
            System.out.println("Thread interrupted: " + e.getMessage());  
        }  
    }  
}  
  
public class Main9 {  
    public static void main(String[] args) {  
        System.out.print("1BM23CS085, Dama Yohitesh Naveen Sai\n");  
  
        Thread thread1 = new DisplayMessage("BMS College of Engineering", 10);  
        Thread thread2 = new DisplayMessage("CSE", 2);  
  
        thread1.start();  
        thread2.start();  
    }  
}
```

OUTPUT:

```
C:\Users\yohit\OneDrive\Desktop\Java LPs>javac Main9.java
C:\Users\yohit\OneDrive\Desktop\Java LPs>java Main9.java
1BM23CS085, Dama Yohitesh Naveen Sai
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
^C
```

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.

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

public class DivisionApp extends JFrame {
    private JTextField num1Field, num2Field, resultField;
    private JButton divideButton;

    public DivisionApp() { // Setting up the frame
        setTitle("Integer Division");
        setSize(300, 200);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLayout(new GridLayout(4, 2));
        // Creating Components
        JLabel num1Label = new JLabel("Num1:");
        num1Field = new JTextField();
        JLabel num2Label = new JLabel("Num2:");
        num2Field = new JTextField();
        JLabel resultLabel = new JLabel("Result:");
        resultField = new JTextField();
        resultField.setEditable(false);
        divideButton = new JButton("Divide");
    }
}
```

```

    add (num1Label);
    add (num1Field);
    add (num2Label);
    add (num2Field);
    add (resultLabel);
    add (resultField);
    add (new JLabel()); // Empty cell
    add (divideButton);

    divideButton.addActionListener (new ActionListener() {
        @Override
        public void actionPerformed (ActionEvent e) {
            try {
                int num1 = Integer.parseInt (num1Field.getText ());
                int num2 = Integer.parseInt (num2Field.getText ());
                if (num2 == 0) {
                    throw new ArithmeticException ("Division by zero");
                }
                int result = num1 / num2;
                resultField.setText (String.valueOf (result));
            } catch (NumberFormatException ex) {
                JOptionPane.showMessageDialog (DivisionApp.this,
                    "Please enter valid integers.");
            } catch (ArithmeticException ex) {
                JOptionPane.showMessageDialog (DivisionApp.this,
                    ex.getMessage (), "Arithmetic Error");
            }
        }
    });
}

```

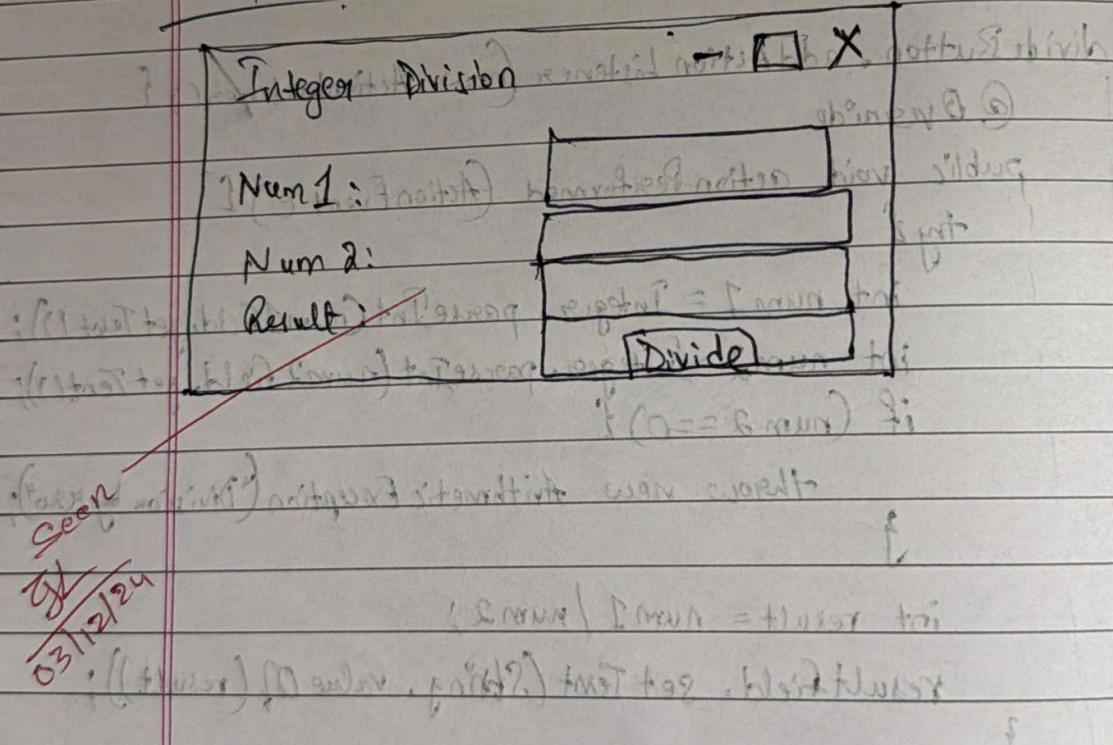
```

public static void main (String [] args) {
    SwingUtilities.invokeLater ( () -> {
        DivisionApp app = new DivisionApp ();
        app.setVisible (true);
    });
}

```

↳ program : (()) start work
↳ (returning division) bbs

Output :



CODE:

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

public class DivisionApp extends JFrame {
    private JTextField num1Field, num2Field, resultField;
    private JButton divideButton;

    public DivisionApp() {
        // Setting up the frame
        setTitle("Integer Division");
        setSize(300, 200);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLayout(new GridLayout(4, 2));

        // Creating components
        JLabel num1Label = new JLabel("Num1:");
        num1Field = new JTextField();
        JLabel num2Label = new JLabel("Num2:");
        num2Field = new JTextField();
        JLabel resultLabel = new JLabel("Result:");
        resultField = new JTextField();
        resultField.setEditable(false);
        divideButton = new JButton("Divide");

        // Adding components to the
        frame add(num1Label);
        add(num1Field);
        add(num2Label)
        ;
        add(num2Field);
        add(resultLabel);
        add(resultField);
        add(new JLabel()); // Empty
        cell add(divideButton);
```

```

// Adding action listener to the button
divideButton.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        try {
            int num1 =
                Integer.parseInt(num1Field.getText()); int num2
                = Integer.parseInt(num2Field.getText());

            if (num2 == 0) {
                throw new ArithmeticException("Division by zero");
            }

            int result = num1 / num2;
            resultField.setText(String.valueOf(result));
        } catch (NumberFormatException ex) {
            JOptionPane.showMessageDialog(DivisionApp.this,
                "Please enter valid integers.", "Number Format Error",
                JOptionPane.ERROR_MESSAGE);
        } catch (ArithmeticException ex) {
            JOptionPane.showMessageDialog(DivisionApp.this,
                ex.getMessage(), "Arithmetic Error",
                JOptionPane.ERROR_MESSAGE);
        }
    }
});

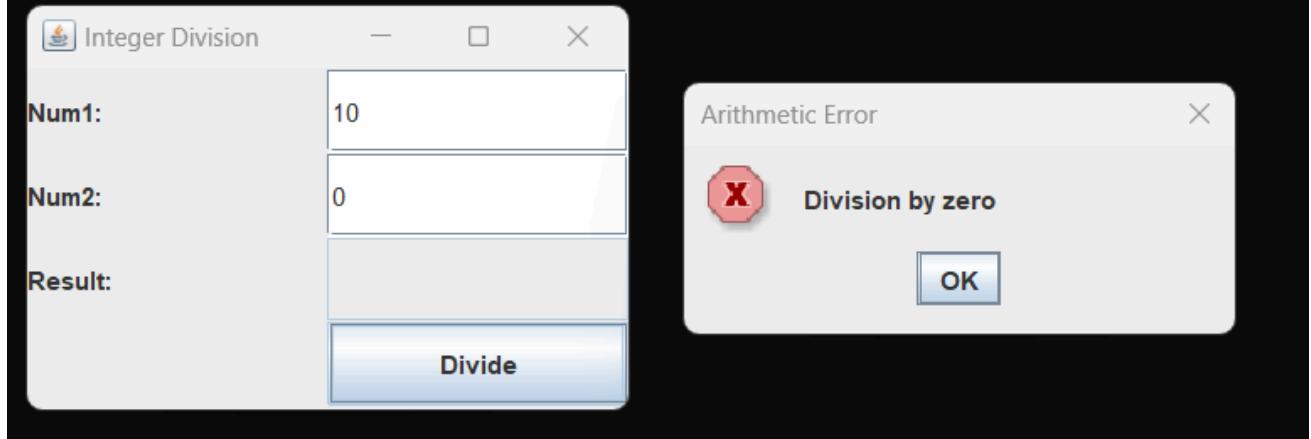
public static void main(String[] args) {
    System.out.print("1BM23CS085, Dama Yohitesh Naveen Sai\n");
    SwingUtilities.invokeLater(() -> {
        DivisionApp app = new DivisionApp();
        app.setVisible(true);
    });
}
}

```

OUTPUT 1:

```
C:\Users\yohit\OneDrive\Desktop\Java LPs>javac DivisionApp.java
```

```
C:\Users\yohit\OneDrive\Desktop\Java LPs>java DivisionApp.java  
1BM23CS085, Dama Yohitesh Naveen Sai
```



OUTPUT 2: