

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



**LAB REPORT**

**on**

**Object Oriented Java Programming**

**(23CS3PCOOJ)**

*Submitted by*

Chitrashree K(**1BM23CS081**)

*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 Chitrashree K(**1BM23CS081**), who is a bonafide student of **B.M.S. College of Engineering**. It is in partial fulfillment for the award of **Bachelor of Engineering in Computer Science and Engineering** of the Visvesvaraya Technological University, Belgaum. The Lab report has been approved as it satisfies the academic requirements in respect of an Object Oriented Java Programming (23CS3PCOOJ) work prescribed for the said degree.

Prof. 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	01/10/24	Quadratic Equations	4-7
2	08/10/24	Calculation of SGPA	8-13
3	15/10/24	Book Information	14-18
4	22/10/24	Finding area using abstract class	19-23
5	29/10/24	Bank class using inheritance concept	24-32
6	12/11/24	Packages	33-40
7	19/11/24	Interfaces	41-45
8	26/11/24	Exception Handling	46-50
9	03/12/24	Thread Programming	51-54
10	03/12/24	Open Ended Exercise	55-60

Github Link:  
<https://github.com/Chitreshree-tech/Java-Programs>

## 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.

LAB PROGRAM - 01	
Q3.	DATE : <p>Develop a Java program that prints all real solutions to the quadratic equation <math>ax^2+bx+c=0</math>. Read in a,b,c and use the quadratic formula. If the discriminant <math>b^2-4ac</math> is -ve, display a message stating that there are no real solutions.</p> <pre>Code: import java.util.*; public class Quadratic {     public static void main (String [] args) {         Scanner scn = new Scanner (System.in);         System.out.println ("Enter 3 nos : ");         double a = scn.nextDouble();         double b = scn.nextDouble();         double c = scn.nextDouble();         double d = (b*b)-(4*a*c);         if (d &gt; 0) {             double root1 = (-b+Math.sqrt(d))/(2*a);             double root2 = (-b-Math.sqrt(d))/(2*a);             System.out.println ("There exists two real solutions.");             System.out.println ("Root1 = " + root1);             System.out.println ("Root2 = " + root2);         }         else if (d == 0) {             double root = -b/(2*a);             System.out.println ("There exist one real solution.");             System.out.println ("Root = " + root);         }         else {             System.out.println ("There exists no real solutions.");         }     } }</pre>

DATE: \_\_\_\_\_

scn. close();

4 or 3  
3

output:

Case 1 : Enter 3 nos:  
1 1 1

There exists no real solution.

Case 2 : Enter 3 nos:  
1 2 1

There exists one real solution

Root = -1.0

Case 3 : Enter 3 nos:  
1 3 1

There exists two real solutions

Root<sub>1</sub> = -0.381966011250105  
Root<sub>2</sub> = -2.618033988749895

~~seen off~~  
~~gt~~  
~~0/10/2014~~

Code:

```
import java.util.*;
public class Quadratic
{
    public static void main(String arg[])
    {
        int a,b,c;
        double d,r1,r2;
        System.out.println("Chitrashree K\n 1BM23CS081");
        Scanner s=new Scanner(System.in);
        System.out.println("enter the value of a:");
        a=s.nextInt();
        System.out.println("enter the value of b:");
        b=s.nextInt();
        System.out.println("enter the value of c:");
        c=s.nextInt();
        d=b*b-(4*a*c);
        if(d==0)
        {
            System.out.println("roots are real and equal");
            r1=r2=-b/(2*a);
            System.out.println("root1=root2="+r1);
        }
        else if(d>0)
        {
            System.out.println("roots are real and distinct");
            r1=(-b+Math.sqrt(d))/(2*a);
            r2=(-b-Math.sqrt(d))/(2*a);
            System.out.println("root1="+r1+"and root2="+r2);
        }
        else
        {
            double real=-b/(2*a);
            double imaginary=Math.sqrt(-d)/(2*a);
            System.out.println("Roots are imaginary:");
            System.out.printf("root1=% .2f+% .2fi and root2=% .2f-% .2fi",real,imaginary,real,imaginary);
        }
    }
}
```

## Output:

```
C:\Users\admin\Desktop\CSK>javac Quadratic.java
C:\Users\admin\Desktop\CSK>java Quadratic
Chitrashree K
1BM23CS081
enter the value of a:
1
enter the value of b:
1
enter the value of c:
1
Roots are imaginary:
root1=0.00+0.87i and root2=0.00-0.87i
C:\Users\admin\Desktop\CSK>javac Quadratic.java

C:\Users\admin\Desktop\CSK>java Quadratic
Chitrashree K
1BM23CS081
enter the value of a:
1
enter the value of b:
2
enter the value of c:
1
roots are real and equal
root1=root2=-1.0

C:\Users\admin\Desktop\CSK>javac Quadratic.java
C:\Users\admin\Desktop\CSK>java Quadratic
Chitrashree K
1BM23CS081
enter the value of a:
1
enter the value of b:
3
enter the value of c:
1
roots are real and distinct
root1=-0.3819660112501051and root2=-2.618033988749895
```

## **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.

08/10/24	LAB PROGRAM - 02	DATE :
Q4.	<p>Develop a Java program to create 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</p> <p>Code:</p> <pre>import java.util.Scanner; class Subject {     int subjectMarks;     int credits;     int grade;     public int calculateGrade() {         if (subjectMarks &lt; 40) {             grade = 0;         } else if (subjectMarks &gt; 100) {             System.out.println("Invalid input.                 Marks cannot be greater than 100");         } else if (subjectMarks ≥ 90) {             grade = 10;         } else if (subjectMarks ≥ 80) {             grade = 9;         } else if (subjectMarks ≥ 70) {             grade = 8;         } else if (subjectMarks ≥ 60) {             grade = 7;         }     } }</pre>	

```

else if (subjectMarks >= 50) {
    grade = 6;
}
else if (subjectMarks >= 40) {
    grade = 5;
}
else {
    grade = 0;
}
return grade;

class Student {
    Scanner a = new Scanner(System.in);
    String USN;
    String Name;
    int credits;
    double[] marks;
    System.out.println("Enter");
    void getDetails() {
        USN = a.nextLine();
        System.out.println("Enter the marks");
        System.out.println("of subject " + i + ": ");
        marks[i] = a.nextDouble();
        credits[i] = a.nextInt();
        System.out.println("Enter the credits");
        System.out.println("of subject " + i + ": ");
        credits[i] = a.nextInt();
    }
    System.out.println();
}

double calculateSGPA() {
    int totalCredits = 0;
    double totalMarks = 0;
    for (int i = 0; i < 8; ++i) {
        totalMarks += (marks[i] * calculateGrade(marks[i]));
        totalCredits += credits[i];
    }
    return totalMarks = 0 ? 0 : totalMarks / totalCredits;
}

```

```

void displayDetails() {
    for (int i=0; i<8; ++i) {
        System.out.println("Name : " + NAME);
        System.out.println("USN : " + USN);
    }
}

public class Student {
    public static void main(String[] args) {
        Student s1 = new Student();
        s1.getDetails();
        s1.displayDetails();
        s1.calculateSGPA();
    }
}

```

~~seen  
process~~

Output:-

Enter student name: AAA  
 Enter student USN :: IBM23LJN08  
 Enter marks for subject1 : 95  
 Enter credits for → : 4  
 Enter marks for subject2: 85  
 Enter credits for → : 3  
 Enter marks for subject3: 94  
 Enter credits → : 4  
 Enter marks for subject4: 84  
 Enter credits → : 3  
 Enter Marks for subject5: 60  
 Enter credits for → : 3  
 Enter marks for subject6: 8  
 Enter credits → : 1  
 Enter marks for subject7: 99  
 Enter credits → : 1  
 Enter marks for subject8: 63  
 Enter credits → : 1

Student Name: AAA

USN: IBM23LJN08

SGPA: 8.60

Code:

```
import java.util.Scanner;

class Subject {
    int subjectMarks;
    int credits;
    int grade;

    public void calculateGrade() {
        if (subjectMarks < 40)
            grade = 0;
        else if (subjectMarks > 100)
            grade = 4;
        else {
            if (subjectMarks >= 90)
                grade = 10;
            else if (subjectMarks >= 80)
                grade = 9;
            else if (subjectMarks >= 70)
                grade = 8;
            else if (subjectMarks >= 60)
                grade = 7;
            else if (subjectMarks >= 50)
                grade = 6;
            else if (subjectMarks >= 40)
                grade = 5;
            else
                grade = -1;
        }
    }
}

class Student1 {
    String name;
    String usn;
    double SGPA;
    Subject subject[];
    Scanner s;
    Student1() {
        subject = new Subject[8];
        for (int i = 0; i < 8; i++) {
            subject[i] = new Subject();
        }
        s = new Scanner(System.in);
    }
}
```

```

}

public void getStudentDetails() {
    System.out.print("Enter student name: ");
    this.name = s.nextLine();
    System.out.print("Enter student USN: ");
    this.usn = s.nextLine();
}

public void getMarks() {
    for (int i = 0; i < 8; i++) {
        System.out.print("Enter marks for subject " + (i + 1) + ": ");
        subject[i].subjectMarks = s.nextInt();
        System.out.print("Enter credits for subject " + (i + 1) + ": ");
        subject[i].credits = s.nextInt();
        subject[i].calculateGrade();
    }
    s.nextLine();
}

public void computeSGPA() {
    double totalPoints = 0;
    int totalCredits = 0;

    for (int i = 0; i < 8; i++) {
        totalPoints += subject[i].grade * subject[i].credits;
        totalCredits += subject[i].credits;
    }

    SGPA = (totalCredits == 0) ? 0 : totalPoints / totalCredits;
}

public void displayResults() {
    System.out.println("Student Name: " + name);
    System.out.println("USN: " + usn);
    System.out.printf("SGPA: %.2f\n", SGPA);
}

public class Student {
    public static void main(String[] args) {
        System.out.println("Chitresh K\\n 1BM23CS081");
        Student1 s1 = new Student1();
        s1.getStudentDetails();
        s1.getMarks();
    }
}

```

```
    s1.computeSGPA();
    s1.displayResults();
}

}
```

**Output:**

```
C:\Users\admin\Desktop\CSK>java Student
Chitrashree K
1BM23CS081
Enter student name: AAA
Enter student USN: 1BM23CS108
Enter marks for subject 1: 95
Enter credits for subject 1: 4
Enter marks for subject 2: 85
Enter credits for subject 2: 3
Enter marks for subject 3: 94
Enter credits for subject 3: 4
Enter marks for subject 4: 84
Enter credits for subject 4: 3
Enter marks for subject 5: 66
Enter credits for subject 5: 3
Enter marks for subject 6: 8
Enter credits for subject 6: 1
Enter marks for subject 7: 99
Enter credits for subject 7: 1
Enter marks for subject 8: 63
Enter credits for subject 8: 1
Student Name: AAA
USN: 1BM23CS108
SGPA: 8.60
```

### **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.

15/10/24 LAB-PROGRAM - D3 PAGE NO : DATE :

Q5 Create a class Book which contains four members: name, author, price, num\_pages. 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.

```
Code: import java.util.*;  
  
public class Book {  
    private String name;  
    private String author;  
    private double price;  
    private int num_pages;  
  
    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 setName(String name) { this.name = name; }  
  
    public void setAuthor(String author) { this.author = author; }  
  
    public void setPrice(double price) { this.price = price; }  
  
    public void setNumPages(int num_pages) { this.num_pages = num_pages; }  
  
    public String getName() { return name; }  
}
```

```
public String getAuthor() {  
    return author;
```

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

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

```
}
```

```
public String toString() {  
    return ("Book Details : In Book  
name: " + name + " In Book Author: " +  
author + " In Book Price: " + price + "  
no. of pages in the book : " + numPages);  
}
```

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

```

① book[i].setName(name);
book[i].setAuthor(author);
book[i].setPrice(price);
book[i].setNumPages(numPages);

book[i] = new Book(name, author, price, numPages);
System.out.println("Entered Book details:");
for (int i=0; i<n; ++i){
    book[i].toString();
}
sc.close();

```

seen execute  
Output - Enter the no of books:

2

Enter book name:

aaa

Enter author name:

dnyuthi

Enter book price:

2000

Enter the no. of pages in the book:

100

Enter book name:

bbb

Enter author name:

chitra

Enter book price:

5000

Enter the no. of pages in the book:

200

Code:

```
import java.util.*;
public class Book{
    private String bname;
    private String author;
    private double price;
    private int num_pages;

    Book(String bname, String author, double price, int num_pages){
        this.bname=bname;
        this.author=author;
        this.price=price;
        this.num_pages=num_pages;
    }
    public void setName(String bname){
        this.bname=bname;
    }
    public void setAuthor(String author){
        this.author=author;
    }
    public void setPrice(double price){
        this.price=price;
    }
    public void setNUmPages(int num_pages){
        this.num_pages=num_pages;
    }

    public String getName(){
        return bname;
    }
    public String getAuthor(){
        return author;
    }
    public double getPrice(){
        return price;
    }
    public int getNumPages(){
        return num_pages;
    }

    public String toString(){
        return ("Book Details:\n Book name:"+bname+"\n Book author:"+author+"\n
Book price:"+price+"\n No.of pages in the book:"+num_pages);
    }
}
```

```

public static void main(String[] args){
    System.out.println("Chitrashree K\n 1BM23CS081"):
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter the no of books:");
        int n=sc.nextInt();
        System.out.println();
        Book[] books=new Book[n];
        sc.nextLine();
        for(int i=0;i<n;++i){
            System.out.println("Enter book name:");
            String bname=sc.nextLine();
            System.out.println("Enter author name:");
            String author=sc.nextLine();
            System.out.println("Enter book price:");
            double price=sc.nextDouble();
            System.out.println("Enter the no of pages in the book:");
            int num_pages=sc.nextInt();
            sc.nextLine();
            books[i]=new Book(bname, author, price, num_pages);
        }
        System.out.println("Entered Book Details are:\n");
        for(int i=0;i<n;i++){
            System.out.println(books[i].toString());
        }
        sc.close();
    }
}

```

## Output:

```

C:\Users\admin\Desktop\CSK>javac Book.java
C:\Users\admin\Desktop\CSK>java Book
Chitrashree K
 1BM23CS081
Enter the no of books:
2

Enter book name:
AAA
Enter author name:
BBB
Enter book price:
5000
Enter the no of pages in the book:
100
Enter book name:
CCC
Enter author name:
DDD
Enter book price:
6000
Enter the no of pages in the book:
150
Entered Book Details are:

Book Details:
Book name:AAA
Book author:BBB
Book price:5000.0
No.of pages in the book:100
Book Details:
Book name:CCC
Book author:DDD
Book price:6000.0
No.of pages in the book:150

```

#### **Program 4**

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

12/10/24 LAB - PROGRAM - 04  
Abstract class - Shape  
3 subclass - Rectangle, Triangle, Circle  
method to print area

code:

```
import java.util.*;  
  
abstract class Shape {  
    public abstract void findArea();  
}  
  
class Rectangle extends Shape {  
    int length; int area;  
    int breadth;  
    Rectangle(int length, int breadth) {  
        this.length = length;  
        this.breadth = breadth;  
    }  
    public void findArea() {  
        area = length * breadth;  
        System.out.println("Area of rectangle  
is " + area);  
    }  
}  
  
class Triangle extends Shape {  
    int base;  
    int height;  
    double area;  
    Triangle(int base, int height) {  
        this.base = base;  
        this.height = height;  
    }  
}
```

```
public void findArea(){  
    area = 0.5 * base * height;  
    System.out.println("Area of triangle  
is " + area);
```

3

```
class Circle extends Shape{  
    int radius;  
    double area;  
    Circle(int radius){  
        this.radius = radius;
```

3

```
        public void findArea(){  
            area = 3.14 * radius * radius;  
            System.out.println("Area of circle  
is " + area);
```

3

```
public class Area{
```

```
    public static void main(String[] args){  
        Scanner a = new Scanner(System.in);  
        System.out.println("To find area of  
rectangle:\n");
```

Rectangle ~~a = new~~

```
        System.out.println("Enter length and  
breadth of the rectangle: ");
```

```
        int l = a.nextInt();
```

```
        int b = a.nextInt();
```

```
        Rectangle r1 = new Rectangle(l, b);  
        r1.findArea();
```

- System.out.println("To find area of triangle");  
- (e: "In");  
System.out.println("Enter base and height of the triangle:");  
int ba = a.nextInt();  
int h = a.nextInt();  
Triangle t = new Triangle(ba, h);  
t.findArea();

2 - System.out.println("To find area of the circle")

System.out.println("Enter the radius of the circle:");  
int ra = a.nextInt();  
Circle c = new Circle(ra);  
c.findArea();

3

~~seen  
execute~~

Output: To find area of rectangle:  
enter length and breadth of rectangle:

2 5

Area of the rectangle is 10

To find area of triangle

enter base and height of triangle:

2 5

Area of the triangle is 5.0

To find area of circle

~~executed~~ enter radius of the circle:

5

Area of the circle is 78.5

~~seen  
execute~~

Code:

```
import java.util.*;  
  
abstract class Shape{  
    public abstract void findArea();  
}  
  
class Rectangle extends Shape{  
    int length, breadth;  
    int area;  
    Rectangle(int length, int breadth){  
        this.length=length;  
        this.breadth=breadth;  
    }  
    public void findArea(){  
        area=length*breadth;  
        System.out.println("Area of the Rectangle is "+area);  
    }  
}  
  
class Triangle extends Shape{  
    int base, height;  
    double area;  
    Triangle(int base, int height){  
        this.base=base;  
        this.height=height;  
    }  
    public void findArea(){  
        area=0.5*base*height;  
        System.out.println("Area of the triangle is "+area);  
    }  
}  
  
class Circle extends Shape{  
    int radius;  
    double area;  
    Circle(int radius){  
        this.radius=radius;  
    }  
    public void findArea(){  
        area=3.14*radius*radius;  
        System.out.println("Area of the circle is "+area);  
    }  
}
```

```

public class Area{
    public static void main(String[] args){
        System.out.println("Chitresh K\n 1BM23CS081");
        Scanner a=new Scanner(System.in);
        System.out.println("To find the area of rectangle:");
        System.out.println("enter length and breadth of the rectangle:");
        int l=a.nextInt();
        int b=a.nextInt();
        Rectangle r=new Rectangle(l, b);
        r.findArea();

        System.out.println("\nTo find the area of triangle:");
        System.out.println("enter base and height of the triangle:");
        int ba=a.nextInt();
        int h=a.nextInt();
        Triangle t=new Triangle(ba, h);
        t.findArea();

        System.out.println("\nTo find the area of circle:");
        System.out.println("enter radius of the circle:");
        int ra=a.nextInt();
        Circle c=new Circle(ra);
        c.findArea();
    }
}

```

Output:

```

C:\Users\admin\Desktop\CSK>java Area
Chitresh K
 1BM23CS081
To find the area of rectangle:
enter length and breadth of the rectangle:
5 2
Area of the Rectangle is 10

To find the area of triangle:
enter base and height of the triangle:
5 2
Area of the triangle is 5.0

To find the area of circle:
enter radius of the circle:
5 2
Area of the circle is 78.5

C:\Users\admin\Desktop\CSK>

```

### **Program 5**

Develop a Java program to create a class Bank that maintains two kinds of account for its customers, one called a savings account and the other current account. The savings account provides

compound interest and withdrawal facilities but no cheque book facility. The current account provides a cheque book facility but no interest. Current account holders should also maintain a minimum balance and if the balance falls below this level, a service charge is imposed.

Create a class Account that stores customer name, account number and type of account. From this

derive the classes Cur-acct and Sav-acct to make them more specific to their requirements.

Include

the necessary methods in order to achieve the following tasks:

- a)Accept deposits from customers and update the balance.
- b)Display the balance.
- c)Compute and deposit interest
- d)Permit withdrawal and update the balance
- e) Check for the minimum balance, impose a penalty if necessary and update the balance.

29/10/21

## LAB - PROGRAM - 05

PAGE NO:  
DATE:

1. Create a class Bank that maintains a account savings, current the savings account provides compound interest and withdrawal facilities but no cheque book facility. current account provides cheque book facility but no interest, also maintain a min balance and if balance falls below this level a service charge is imposed.
- create class Account → customer\_name, acc\_no
- Derive classes cur-acct & sav-acct. methods
- a) accept deposit & update balance.
  - b) display the balance.
  - c) compute and deposit interest.
  - d) Permit withdrawal & update balance.
  - Check for min balance, impose penalty if necessary & update balance.

Code:

```
abstract class Account {
    String customer_name;
    String account_no;
    String account_type;
    double balance = 0.0;

    Account(String customer_name, String account_no, String account_type) {
        this.customer_name = customer_name;
        this.account_no = account_no;
        this.account_type = account_type;
    }

    public void display_bal() {
        System.out.println("Balance is " + balance);
    }

    public void deposit(double d_amount) {
        balance += d_amount;
        System.out.println("Deposited: " + d_amount);
        display_bal();
    }
}
```

abstract void withdraw(double w-amount);

{

class CurrAcc extends Account {

final double min-bal = 500.00;

final double penalty = 50.0;

super(customer-name, account-no, account-type);

public void checkBook(String to-name,  
double c.amount) {

System.out.println("Check issued to :" +  
to-name);

balance -= c.amount;

display-bal(); minBalance();

{

public void minBalance() {

if (balance < min-bal) {

balance -= penalty;

System.out.println("Penalty applied");

public void withdraw(  
double amount);

if (amount > balance) {

System.out.println("No  
sufficient balance");

else {

balance -= amount;

display-bal(); minBalance();

class SavAcc extends Account {

final double interest-rate = 7.0;

SavAcc super(customer-name, account-no, account-type);

public void computeInterest() {

double interest = balance \* interest-rate / 100;

balance += interest;

display-bal();

public void withdraw(double amount) {

if (amount > balance) {

System.out.println("No sufficient bal-  
ance");

```

else {
    balance -= amount;
}
display_bal();
}

public class Bank {
    public static void main (String [] args) {
        Scanner a = new Scanner(System.in);
        System.out.println ("Enter your name: ");
        String customer_name = a.nextLine();
        System.out.println ("Enter account number: ");
        String account_no = a.nextLine();
        System.out.println ("Enter account type  

(savings / current): ");
        String account_type = a.nextLine().toLowerCase()
                               .case();
        if (account_type == "savings") {
            SavAcc account = new SavAcc (customer_
name, account_no, account_type);
        } else {
            CurrAcc account = new CurrAcc (customer_
name, account_no, account_type);
        }
        while (true) {
            int choice;
            System.out.println ("1. Deposit\n2. Display  

3. Compute Interest\n4. Withdrawal  

5. Exit");
            System.out.print ("Enter your choice: ");
            choice = a.nextInt();
            switch (choice) {
                case 1:
                    System.out.println ("Enter amount: ");
                    double d_amount = a.nextDouble();
                    account.deposit (d_amount); break;
            }
        }
    }
}

```

case 2:

account.display(); break;

case 3:

```
if (account.type == "savings") {  
    account.computeInterest(); }  
else {
```

```
    System.out.println("Enter person name");  
    String name = a.nextLine();  
    System.out.println("Enter amount.");  
    double amount = a.nextDouble();  
    account.checkBook(name, amount);
```

case 4:

```
System.out.println("Enter amount.");  
double amount = a.nextDouble();  
amount.withdraw(amount);
```

case 5:

exit();

default:

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

3  
3  
3  
3

~~seen~~

Output:- Enter your name: chitra

Enter ac no: 84123467

Enter ac type (savings/current):  
current

1. Deposit 2. Display 3. Compute /check 4. withdraw [exit]  
1. Enter amount: 2575

Balance is 2575

2. Balance is 2575

3. Enter person name: dyuthi; amount 500  
Check issued to dyuthi

4. withdraw: 2000

Balance: 575

~~executed~~  
~~10/10/14~~

Code:

```
import java.util.*;  
  
abstract class Account {  
    String customer_name, account_no, account_type;  
    double balance = 0.0;  
  
    Account(String customer_name, String account_no, String account_type) {  
        this.customer_name = customer_name;  
        this.account_no = account_no;  
        this.account_type = account_type;  
    }  
  
    public void display() {  
        System.out.println("Balance is: " + balance);  
    }  
  
    public void deposit(double d_amount) {  
        balance += d_amount;  
        System.out.println("Deposited: " + d_amount);  
        display();  
    }  
  
    abstract void withdraw(double w_amount);  
}  
  
class CurrAcc extends Account {  
    final double min_bal = 500.0;  
    final double penalty = 50.0;  
  
    CurrAcc(String customer_name, String account_no, String account_type) {  
        super(customer_name, account_no, account_type);  
    }  
  
    public void minbalance() {  
        if (balance < min_bal) {  
            balance -= penalty;  
            System.out.println("Penalty Applied");  
        }  
        display();  
    }  
  
    public void checkbook(String to_name, double c_amount) {  
        System.out.println("Check issued to " + to_name);  
        balance -= c_amount;  
        minbalance();  
    }  
}
```

```

}

public void withdraw(double amount) {
    if (amount > balance) {
        System.out.println("No sufficient balance");
    } else {
        balance -= amount;
    }
    minbalance();
}

class SavAcc extends Account {
    final double interest_rate = 7.0;

    SavAcc(String customer_name, String account_no, String account_type) {
        super(customer_name, account_no, account_type);
    }

    public void computeInterest() {
        double interest = balance * interest_rate / 100;
        balance += interest;
        display();
    }

    public void withdraw(double amount) {
        if (amount > balance) {
            System.out.println("No sufficient balance");
        } else {
            balance -= amount;
        }
        display();
    }
}

public class Bank {
    public static void main(String[] args) {
        System.out.println("Chitreshree K\n 1BM23CS081");
        Scanner a = new Scanner(System.in);
        System.out.println("Enter your name:");
        String name = a.nextLine();
        System.out.println("Enter acc no:");
        String acc_no = a.nextLine();
        System.out.println("Enter acc type (savings/current):");
        String acc_type = a.nextLine().toLowerCase();
        Account account;
    }
}

```



## Output:

```
C:\Users\admin\Desktop\CSK>java Bank
Chitrashree K
1BM23CS081
Enter your name:
AAA
Enter acc no:
123456
Enter acc type (savings/current):
savings
1.Deposit.
2.Display.
3.Compute Interest/Check Book.
4.Withdrawal.
5.Exit
1
Enter amount:
5000
Deposited: 5000.0
Balance is: 5000.0
1.Deposit.
2.Display.
3.Compute Interest/Check Book.
4.Withdrawal.
5.Exit
3
Balance is: 5350.0
1.Deposit.
2.Display.
3.Compute Interest/Check Book.
4.Withdrawal.
5.Exit
2
Balance is: 5350.0
1.Deposit.
2.Display.
3.Compute Interest/Check Book.
4.Withdrawal.
5.Exit
```

## **Program 6**

Create a package CIE which has two classes- Student and Internals. The class Student has members like usn, name, sem. The class Internals derived from Student 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.

12/11/2021 LAB PROGRAM - 06

PAGE NO: \_\_\_\_\_  
DATE: \_\_\_\_\_

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.

```
cie - Student: package cie  
import java.util.*;  
public class Student {  
    String usn;  
    String name;  
    int sem; } public void studentDetails(){  
Scanner a = new Scanner(System.in);  
System.out.println("Enter student details:");  
System.out.println("Enter usn:");  
usn = a.nextLine();  
System.out.println("Enter name:");  
name = a.nextLine();  
System.out.println("Enter sem:");  
sem = a.nextInt();  
}  
public void displayStudentDetails(){  
    System.out.println("Name: " + name);  
    System.out.println("USN: " + usn);  
    System.out.println("Sem: " + sem);  
}
```

3 4

cie  
internals

```
package CIE;
import java.util.*;
public class Internals extends Student {
    int [ ] marks = new int [5];
    public void cieMarks() {
        Scanner a = new Scanner (System.in);
        System.out.println ("Enter internal marks");
        for (int i=0; i<5; i++) {
            System.out.println ("Enter marks for
course " + (i+1) + ":");
            marks [i] = a.nextInt();
        }
    }
    public void displaycieMarks() {
        System.out.println ("Internal marks:");
        for (int i=0; i<5; i++) {
            System.out.println ("course " + (i+1) + ":"
marks [i]);
        }
    }
}
```

SBE  
Externals

```
import CIE.Internals;
import java.util.*;
public class External extends Internals {
    int [ ] emarks = new int [5];
    int [ ] total = new int [5];
    Scanner a = new scanner (System.in);
    public void seeMarks() {
        System.out.println ("Enter external marks");
        for (int i=0; i<5; i++) {
            System.out.println ("Enter marks for
course " + (i+1) + ":");
            emarks [i] = a.nextInt();
        }
    }
}
```

```
public void calTotalMarks() {
    for (int i=0; i<5; ++i) {
        total[i] = marks[i] + eMarks[i];
    }
}
```

```
public void displayFinalMarks() {
    System.out.println("Display Student Details");
    System.out.println("External Marks:");
    for (int i=0; i<5; ++i) {
        System.out.println("External Marks:");
        Course[i+1] += externalMarks[i];
    }
}
System.out.println("Final Marks:");
for (int i=0; i<5; ++i) {
    System.out.println("Course " + (i+1) + ":" + finalMarks[i]);
}
}
```

```
main
import packages.SEE.external;
import java.util.Scanner;
public class Main {
    public static void main (String [] args) {
        Scanner sc = new Scanner (System.in);
        System.out.println ("Enter the number of students:");
        int n = sc.nextInt();
        sc.nextLine();
        External [] students = new External [n];
        for (int i=0; i<n; ++i) {
            students[i] = new External();
            System.out.println ("Enter details for student " + (i+1));
        }
    }
}
```

students[i].inputStudentDetails();  
 students[i].inputCI\_marks();  
 students[i].inputSE\_marks();

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

students[i].calcTotalMarks();

students[i].displayFinalMarks();

}  
 sc.close();

~~dp see~~

Output:- Enter the no. of students :)

Enter details for student

Enter USN: IBM238000

Enter Name: aaaa

Enter Semester: 1

Enter marks for course 1: 96

                1      2: 95

                0      3: 98

                1      4: 92

                2      5: 95

Enter External Marks for 5 courses:

Enter marks for course 1: 98

                1      2: 99

                1      3: 99

                1      4: 97

                1      5: 96

Student Details:

USN: IBM238000

Name: aaaa

Semester: 1

Internal marks:

Course 1: 96

Course 2: 95

Course 3: 98

Course 4: 92

Course 5: 95

External Marks:

Course 1: 98

Course 2: 99

Course 3: 99

Course 4: 97

Course 5: 96

Final Marks:

Course 1: 194

Course 2: 194

Course 3: 197

Course 4: 189

Course 5: 190

Code:

**Main class:**

```
import SEE.External;
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the number of students: ");
        int n = sc.nextInt();
        sc.nextLine(); // Consume the newline character
        External[] students = new External[n];

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

        sc.close();
    }
}
```

**In CIE package:**

```
package CIE;
```

```
import java.util.Scanner;

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

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

```

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

package CIE;

import java.util.Scanner;

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

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

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

```

**In SEE Package:**

```

package SEE;

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

public class External extends Internals {
    protected int[] externalMarks = new int[5];
    protected int[] finalMarks = new int[5];

    public void inputSEEmarks() {
        Scanner s = new Scanner(System.in);
        System.out.println("Enter External Marks for 5 Courses: ");
    }
}

```

```

for (int i = 0; i < 5; i++) {
    System.out.println("Enter marks for course " + (i + 1) + ": ");
    externalMarks[i] = s.nextInt();
}

public void calculateFinalMarks() {
    for (int i = 0; i < 5; i++) {
        finalMarks[i] = marks[i] + externalMarks[i];
    }
}

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

```

## Output:

```

USN: 1BM23CS081
Name: Chitrashree K
Semester: 2
Internal Marks:
Course 1: 90
Course 2: 91
Course 3: 92
Course 4: 93
Course 5: 94
External Marks:
Course 1: 95
Course 2: 96
Course 3: 97
Course 4: 98
Course 5: 99
Final Marks:
Course 1: 185
Course 2: 187
Course 3: 189
Course 4: 191
Course 5: 193
USN: 1BM23CS081
Name: Chitrashree K
Semester: 3
Internal Marks:
Course 1: 89
Course 2: 88
Course 3: 87
Course 4: 86
Course 5: 85
External Marks:
Course 1: 84
Course 2: 83
Course 3: 2
Course 4: 81
Course 5: 80
Final Marks:
Course 1: 173
Course 2: 171
Course 3: 89
Course 4: 167
Course 5: 165

```

## Program 7

### Interface Program

19/11/24 LAB PROGRAM-D7  
Interface Program:

code

```
interface Polygon {
    default double getPerimeter(){
        return 0.0;
    }
    abstract double getArea();
}

class Triangle implements Polygon {
    double x, y, z;
    double area;
    Triangle(double x, double y, double z) {
        this.x = x;
        this.y = y;
        this.z = z;
    }
    public double getPerimeter() {
        return x + y + z;
    }
    public double getArea() {
        double s = (x + y + z) / 2.0;
        return Math.sqrt(s * (s - x) * (s - y) * (s - z));
    }
}

class Rectangle implements Polygon {
    double x, y;
    Rectangle(double x, double y) {
        this.x = x;
        this.y = y;
    }
    public double getPerimeter() {
        return (x + y) * 2;
    }
}
```

```
public double getArea() {  
    return x * y;
```

}

```
class Circle implements Polygon {
```

```
    double r;
```

```
    Circle(double r) {
```

```
        this.r = r;
```

}

```
    public double getPerimeter() {
```

```
        return 2 * 3.14 * r;
```

}

```
    public double getArea() {
```

```
        return 3.14 * r * r;
```

}

```
public class Perimeter {
```

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

```
        Polygon triangle = new Triangle(5.5, 3.5, 4.5);
```

```
        Polygon rectangle = new Rectangle(5.5, 4.5);
```

```
        Polygon circle = new Circle(5.5);
```

```
        System.out.println("Rectangle Area: " +
```

```
            rectangle.getArea());
```

```
        System.out.println("Rectangle Perimeter: " +
```

```
            rectangle.getPerimeter());
```

```
        System.out.println("Circle Area: " + circle.getArea());
```

```
        System.out.println("Circle Perimeter: " + circle.getPerimeter());
```

```
        System.out.println("Triangle Area: " + triangle.
```

```
            .getArea() + " In Triangle Perimeter: " + triangle.
```

```
            .getPerimeter());
```

}

}

Output:-

Rectangle Area: 21.75  
→ Perimeter: 20.0

Circle Area: 94.985

→ Perimeter: 31.54

Triangle Area: 7.854

→ Perimeter: 13.5

Cool

100/  
19/12/2015

Code :

```
interface Polygon{
    default double getPerimeter(){
        return 0.0;
    }
    abstract double getArea();
}

class Triangle implements Polygon{
    double x,y,z;
    double area;
    Triangle(double x,double y, double z){
        this.x=x;
        this.y=y;
        this.z=z;
    }
    public double getPerimeter(){
        return x+y+z;
    }
    public double getArea(){
        double s=(x+y+z)/2;
        area=Math.sqrt(s*(s-x)*(s-y)*(s-z));
        return area;
    }
}
class Rectangle implements Polygon{
    double x,y;
    double area;
    Rectangle(double x,double y){
        this.x=x;
        this.y=y;
    }
    public double getPerimeter(){
        return (2*x)+(2*y);
    }
    public double getArea(){
        area=x*y;
        return area;
    }
}
class Circle implements Polygon{
    double r;
    double area;
    Circle(double r){
        this.r=r;
    }
    public double getPerimeter(){
```

```

        return 2*3.14*r;
    }
    public double getArea(){
        area=3.14*r*r;
        return area;
    }
}
public class Perimeter{
    public static void main(String[] args) {
        Polygon triangle = new Triangle(5.5, 3.5, 4.5);
        Polygon rectangle = new Rectangle(5.5, 4.5);
        Polygon circle = new Circle(5.5);
        System.out.println("Chitreshree K\n 1BM23CS081");
        System.out.println("Rectangle Area: " + rectangle.getArea());
        System.out.println("Rectangle Perimeter: " + rectangle.getPerimeter());

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

        System.out.println("Triangle Area: " + triangle.getArea());
        System.out.println("Triangle Perimeter: " + triangle.getPerimeter());
    }
}

```

## Output

```

C:\Users\Admin\Desktop\1BM23CS081>javac Perimeter.java
C:\Users\Admin\Desktop\1BM23CS081>java Perimeter
Chitreshree K
 1BM23CS081
Rectangle Area: 24.75
Rectangle Perimeter: 20.0
Circle Area: 94.985
Circle Perimeter: 34.54
Triangle Area: 7.854885024620029
Triangle Perimeter: 13.5

```

### **Program 8**

Write a program that demonstrates handling of exceptions in inheritance trees. Create a base class called "Father" and a derived class called "Son" which extends the base class. In Father class, implement a constructor which takes the age and throws the exception WrongAge( ) when the input age<0. In Son class, implement a constructor that causes both father and son's age and throws an exception if son's age is >=father's age.

#### - LAB - PROGRAM - 08

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  $\geq$  father's age

```
import java.util.Scanner;
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 ("Father's age cannot be negative");
        }
    }
}
```

this. fage = fage;

3  
class Son extends Father {  
 int sage;  
 public Son(int fage, int sage) throws  
 WrongAgeException {  
 super(fage);  
 if (sage >= 0) {  
 throw new WrongAgeException ("Son's  
 age cannot be negative.");  
 }  
 if (sage > fage) {  
 throw new WrongAgeException ("Son's age  
 cannot be ≥ father's age.");  
 }  
 }  
}

this. sage = sage;

3  
public class exception\_inheritance {  
 public static void main (String [] args) {  
 try {  
 Father father = new Father(45);  
 Son son = new Son(45, 18);  
 System.out.println ("Father's age: " +  
 father.fage + " Son's age: " + son.sage);  
 Father father = new Father(-45);  
 catch (WrongAgeException e) {  
 System.out.println ("Exception caught:  
 " + e.getMessage());  
 }  
 }  
 }  
}

try {  
 son = son + newSon(10, 50);

} catch (WrongAgeException e) {

System.out.println("Exception caught  
 " + e.getMessage());

}

}

Output: Father's age: 45;

Sons' age: 18

exception caught: Father's age cannot be negative

exception caught: Son's age cannot be greater than / equal to father's age.

Seen

26/1/24

Code:

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

class Father {
    int fage;
    public Father(int fage) throws WrongAgeException {
        if (fage < 0) {
            throw new WrongAgeException("Father's age cannot be negative.");
        }
        this.fage = fage;
    }
}
class Son extends Father {
    int sage;
    public Son(int fage, int sage) throws WrongAgeException {
        super(fage);
        if (sage < 0) {
            throw new WrongAgeException("Son's age cannot be negative.");
        }
        if (sage >= fage) {
            throw new WrongAgeException("Son's age cannot be greater than or equal to father's
age.");
        }
        this.sage = sage;
    }
}
public class ExceptionInheritanceDemo {
    public static void main(String[] args) {
        try {
            Father father = new Father(45);
            Son son = new Son(45, 18);
            System.out.println("Father's age: " + father.fage);
            System.out.println("Son's age: " + son.sage);
            Father father = new Father(-45);
        } catch (WrongAgeException e) {
            System.out.println("Exception caught: " + e.getMessage());
        }
        try {
            Son son = new Son(40, 50);
        } catch (WrongAgeException e) {
```

```
        System.out.println("Exception caught: " + e.getMessage());
    }
}
}
```

**Output:**

```
C:\Users\Admin\Desktop\1BM23CS081>javac ExceptionInheritanceDemo.java
C:\Users\Admin\Desktop\1BM23CS081>java ExceptionInheritanceDemo
Father's age: 45
Son's age: 18
Exception caught: Father's age cannot be negative.
Exception caught: Son's age cannot be greater than or equal to father's age.
```

## **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.

31/12/24 LAB - PROGRAM - 09  
PAGE NO :  
DATE :

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.

```
public class DisplayTwoThreads {
    class DisplayBMSThread extends Thread {
        public void run() {
            try {
                while (true) {
                    System.out.println("BMS College");
                    Thread.sleep(10000);
                }
            } catch (InterruptedException e) {
                System.out.println("Thread interrupted");
            }
        }
    }

    class DisplayCSThread extends Thread {
        public void run() {
            try {
                while (true) {
                    System.out.println("CSE");
                    Thread.sleep(2000);
                }
            } catch (InterruptedException e) {
                System.out.println("Thread interrupted");
            }
        }
    }
}
```

```
public static void main(String[] args){  
    Thread bms = new DisplayBMSThread();  
    Thread cse = new DisplayCSEThread();  
    bms.start();  
    cse.start();
```

{

}

Output:- BMS College

CSE

CSE

CSE

CSE

CSE

BMS College

CSE

seen

Code:

```
public class TwoThreads{

    static class DisplayBMSCollegeThread extends Thread {
        public void run() {
            try {
                while (true) {
                    System.out.println("BMS College of Engineering");
                    Thread.sleep(10000);
                }
            } catch (InterruptedException e) {
                System.out.println("Thread Interrupted");
            }
        }
    }

    static class DisplayCSEThread extends Thread {
        public void run() {
            try {
                while (true) {
                    System.out.println("CSE");
                    Thread.sleep(2000);
                }
            } catch (InterruptedException e) {
                System.out.println("Thread Interrupted");
            }
        }
    }

    public static void main(String[] args) {
        Thread bmsThread = new DisplayBMSCollegeThread();
        Thread cseThread = new DisplayCSEThread();
        bmsThread.start();
        cseThread.start();
    }
}
```

Output:

```
C:\Users\Admin\Desktop\1BM23CS081>java TwoThreads
Chitrashree K
1BM23CS081
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
```

### **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.

3/12/2021 LAB PROGRAM - 10 PAGE NO : DATE :

Write a program that creates a user interface to perform integer divisions. The user enters two numbers in the text fields, Num1 & Num2. The division of Num1 & Num2 is displayed in the Result field when the Divide button is clicked. If Num1 & 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.

Code.

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
public class IntegerDivision GUI {
    public static void main(String[] args) {
        JFrame frame = new JFrame("Integer Division");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setSize(600, 500);
        JLabel label1 = new JLabel("Num1:");
        JLabel label2 = new JLabel("Num2:");
        JTextField num1Field = new JTextField(10);
        JTextField num2Field = new JTextField(10);
        JButton divideButton = new JButton("Divide");
        JLabel resultLabel = new JLabel("Result:");
        frame.setLayout(new FlowLayout());
        frame.add(label1);
        frame.add(num1Field);
        frame.add(label2);
        frame.add(num2Field);
        frame.add(divideButton);
        frame.add(resultLabel);
    }
}
```

```
frame.add(divideButton);
frame.add(resultLabel);

divideButton.addActionListener(new ActionListener{
    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;
            resultLabel.setText("Result: " + result);
        } catch (NumberFormatException ex) {
            JOptionPane.showMessageDialog(frame,
                "Invalid input. Please enter integers",
                "Error", JOptionPane.ERROR_MESSAGE);
        } catch (ArithmeticException ex) {
            JOptionPane.showMessageDialog(frame,
                ex.getMessage(), "Error", JOptionPane.ERROR_MESSAGE);
        }
    }
});

frame.setVisible(true);
```

Output: ① Integer Division

Num1: [81] Num2: [9] Divide]

Result: 9

② Integer Division

Num1: [50.0] Num2: [6.9] Divide]

Result:

~~error  
[X] Invalid input. Please enter integers.~~

③ Integer Division

Num1: [5] Num2: [0] Divide]

Result:

~~error  
[X] Division by zero~~

seen

~~gl  
03/12/24~~

Code:

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

class SwingDemo {
    SwingDemo() {
        // Create JFrame container
        JFrame jfrm = new JFrame("Divider App");
        jfrm.setSize(275, 150);
        jfrm.setLayout(new FlowLayout());

        // To terminate on close
        jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        // Text label
        JLabel jlab = new JLabel("Enter the divisor and dividend:");

        // Add text fields for both numbers
        JTextField ajtf = new JTextField(8);
        JTextField bjtf = new JTextField(8);

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

        // Labels for displaying error and result
        JLabel err = new JLabel();
        JLabel alab = new JLabel();
        JLabel blab = new JLabel();
        JLabel anslab = new JLabel();

        // Add components in order
        jfrm.add(err); // to display error messages
        jfrm.add(jlab);
        jfrm.add(ajtf);
        jfrm.add(bjtf);
        jfrm.add(button);
        jfrm.add(alab);
        jfrm.add(blab);
        jfrm.add(anslab);

        // Button action listener
        button.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent evt) {
                try {
```

```

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

        alab.setText("A = " + a);
        blab.setText("B = " + b);
        anslab.setText("Ans = " + ans);
        err.setText(""); // Clear any previous errors
    } catch (NumberFormatException e) {
        alab.setText("");
        blab.setText("");
        anslab.setText("");
        err.setText("Enter Only Integers!");
    } catch (ArithmaticException e) {
        alab.setText("");
        blab.setText("");
        anslab.setText("");
        err.setText("B should be NON zero!");
    }
}
});
jfrm.setVisible(true);
}

public static void main(String args[]) {

    SwingUtilities.invokeLater(new Runnable() {
        public void run() {
            new SwingDemo();
        }
    });
}
}

```

Output:

The screenshot shows a Java Swing application window titled "Integer Division". The window has a standard title bar with minimize, maximize, and close buttons. Below the title bar is a toolbar with three buttons: a minus sign, a square, and an X. The main area contains four text input fields labeled "Num1:", "Num2:", "Divide", and "Result". The "Divide" button is blue and appears to be the primary action button. To the right of the "Result" field is a vertical scroll bar. The scroll bar has some faint text visible on its track, including "ss" and "ds".

