

**VISVESVARAYA TECHNOLOGICAL
UNIVERSITY, JNANASANGAMA,
BELGAUM - 590014, KARNATAKA**



LABORATORY RECORD
ON
Object Oriented Java Programming
(23CS3PCOOJ) Submitted by
AKSHAY RAJ ARYAN (1BM22CS032)

In partial fulfilment 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
December-2022 to April-2023

INDEX

| Sl. No. | Date | Experiment Title | Page No. |
|----------------|-------------|---|-----------------|
| 1 | 18.12.2023 | Laboratory Program – 1 | 3-9 |
| 2 | 01.01.2024 | Laboratory Program – 2 | 10-15 |
| 3 | 01.01.2024 | Laboratory Program – 3 | 16-20 |
| 4 | 08.01.2024 | Laboratory Program – 4 | 21-25 |
| 5 | 08.01.2024 | Laboratory Program – 5 | 26-34 |
| 6 | 22.01.2024 | Laboratory Program – 6 | 36-42 |
| 7 | 22.01.2024 | Laboratory Program – 7 | 43-47 |
| 8 | 05.02.2024 | Laboratory Program – 8 | 48-52 |
| 9 | 19.02.2024 | Laboratory Program – 9 (And Report on few AWT program) | 53-60 |

LABORATORY PROGRAM - 1

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

```
import java.util.Scanner;
import java.lang.Math;
class Quad
{
    double Disc(double a,double b,double c)
    {
        return b*b-4*a*c;
    }
    void roots(double a,double b, double c)
    {
        double D = Disc(a,b,c);

        if (D<0)
        {
            double realPart = -b/(2*a);
            double imaginaryPart = Math.sqrt(Math.abs(D))/(2*a);
            System.out.println("The Quadratic Equation has Conjugate Imaginary
roots:");
            System.out.printf("Root 1: %.5f + %.5fi\n",realPart,imaginaryPart);
            System.out.printf("Root 2: %.5f - %.5fi\n",realPart,imaginaryPart);
        }
        else if (D>0)
        {
            System.out.println("The Quadratic Equation has Two Distinct Real Roots:");
            double r1=(-b+Math.sqrt(D))/(2*a);
            double r2=(-b-Math.sqrt(D))/(2*a);
            System.out.printf("Root 1: %.5f\n",r1);
            System.out.printf("Root 2: %.5f\n", r2);
        }
        else
        {
            System.out.println("The Quadratic Equation has Equal and Real Root:");
            double r1=(-b)/(2*a);
            System.out.printf("Both Root 1 and Root 2: %.5f\n",r1);
        }
    }
}

class QuadEqn
{
    public static void main(String sx[])
}
```

```

{
    Scanner S1 = new Scanner(System.in);
    System.out.println("Enter the Coefficients of Quadratic Equation :");
    double a = S1.nextDouble();
    double b = S1.nextDouble();
    double c = S1.nextDouble();
    if (a==0)
    {
        System.out.println("Since the Coefficient of x^2 is Zero, it's not a Quadratic
                           Equation");
    }
    else
    {
        Quad quadratic=new Quad();
        quadratic.Disc(a,b,c);
        quadratic.roots(a,b,c);
    }
}
}

```

OUTPUT

```

D:\NotePad++\Java>javac QuadEqn.java

D:\NotePad++\Java>java QuadEqn
Enter the Coefficients of Quadratic Equation :
23
24
26
The Quadratic Equation has Conjugate Imaginary roots:
Root 1: -0.52174 + 0.92640i
Root 2: -0.52174 - 0.92640i

D:\NotePad++\Java>java QuadEqn
Enter the Coefficients of Quadratic Equation :
1
2
1
The Quadratic Equation has Equal and Real Root:
Both Root 1 and Root 2: -1.00000

D:\NotePad++\Java>java QuadEqn
Enter the Coefficients of Quadratic Equation :
6
8
0
The Quadratic Equation has Two Distinct Real Roots:
Root 1: 0.00000
Root 2: -1.33333

```

```


    if
    r1 = (-b) / (2 * a);
    system.out.println("Roots are real and equal");
    system.out.println("Root1 = Root2 = " + r1);
    else if (D > 0)
    {
        r1 = ((-b) + (math.sqrt(a))) / (double)(2 * a);
        r2 = ((-b) - (math.sqrt(a))) / (double)(2 * a);
        system.out.println("Roots are real and different");
        system.out.println("Root1 = " + r1 + "Root2 = " + r2);
    }
    else (D < 0)
    {
        system.out.println("Roots are imaginary and real roots do not exist");
    }
}

class QuadraticEq
{
    public static void main (String xx[])
    {
        Quadratic q = new Quadratic ();
        q.getd();
        q.compute();
    }
}


```

OUTPUT

Enter the coefficients of quadratic equation

23

24

26

The quadratic equation has imaginary roots.

Enter the coefficients.

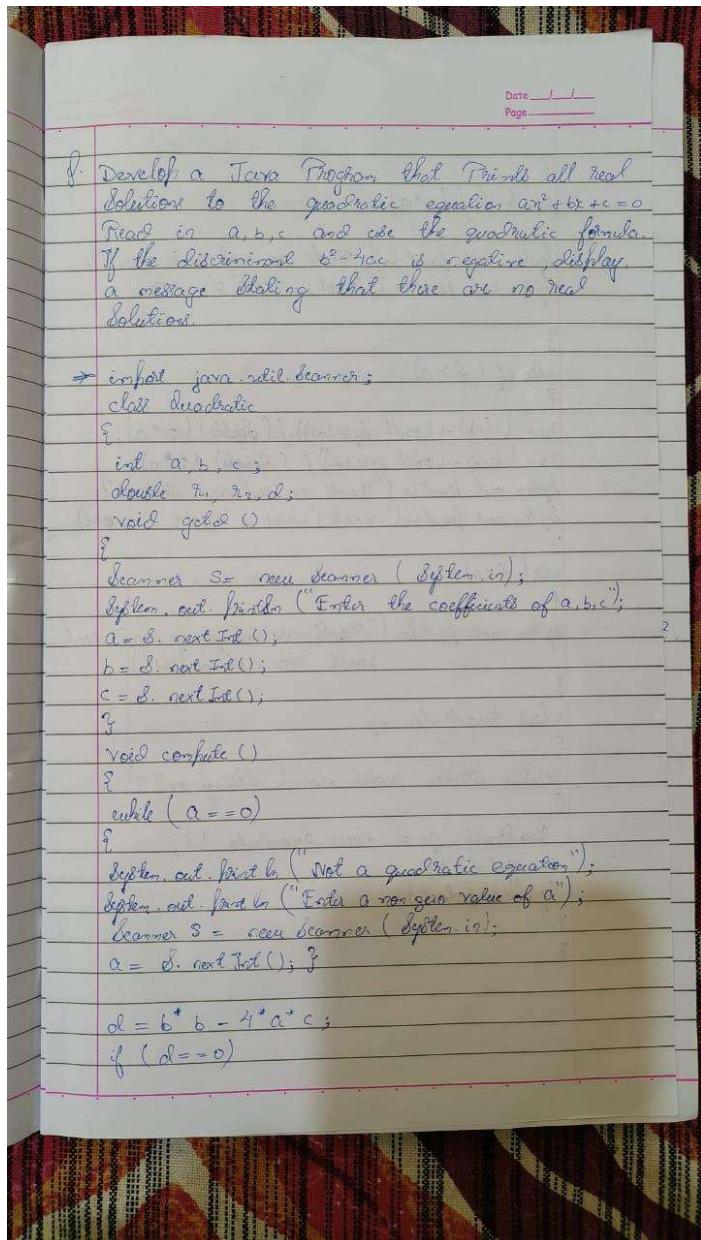
1

2

1

Root = -1.00

The quadratic equation has real and equal root.



LABORATORY PROGRAM - 2

Develop a Java program to create a class Student with members usn, name, an array credits and an array marks. Include methods to accept and display details and a method to calculate SGPA of a student.

```
import java.util.Scanner;

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

    public void acceptDetails() {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter USN: ");
        usn = sc.nextLine();
        System.out.println("Enter Name: ");
        name = sc.nextLine();
        System.out.println("Enter number of subjects: ");
        int n = sc.nextInt();
        credits = new int[n];
        marks = new int[n];
        for (int i = 0; i < n; i++) {
            System.out.println("Enter credits for subject " + (i + 1) + ": ");
            credits[i] = sc.nextInt();
            System.out.println("Enter marks for subject " + (i + 1) + ": ");
            marks[i] = sc.nextInt();
        }
    }

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

    public double calculateSGPA() {
```

```
double totalGrade = 0;
int totalCredit = 0;
for (int i = 0; i < credits.length; i++) {
    totalGrade += getGrade(marks[i]) * credits[i];
    totalCredit += credits[i];
}
return totalGrade / totalCredit;
}

private double getGrade(int marks) {
    if (marks >= 90) {
        return 10;
    } else if (marks >= 80) {
        return 9;
    } else if (marks >= 70) {
        return 8;
    } else if (marks >= 60) {
        return 7;
    } else if (marks >= 50) {
        return 6;
    } else if (marks >= 40) {
        return 5;
    } else {
        return 0;
    }
}

public static void main(String[] args) {
    Student student = new Student();
    student.acceptDetails();
    student.displayDetails();
    System.out.println("SGPA: " + student.calculateSGPA());
}
}
```

OUTPUT

```
D:\NotePad++\Java>javac Student.java

D:\NotePad++\Java>java Student
Enter USN:
L24
Enter Name:
Royce
Enter number of subjects:
3
Enter credits for subject 1:
4
Enter marks for subject 1:
97
Enter credits for subject 2:
3
Enter marks for subject 2:
98
Enter credits for subject 3:
2
Enter marks for subject 3:
99
USN: L24
Name: Royce
Marks:
Subject 1: 97
Subject 2: 98
Subject 3: 99
Credits:
Subject 1: 4
Subject 2: 3
Subject 3: 2
SGPA: 10.0
```

LAB 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;

class Student

{

String USN;

String name;

int [] credits;

int [] marks;

Public void acceptDetails () {

Scanner sc = new Scanner (System.in);

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

USN = sc.nextLine();

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

name = sc.nextLine();

credits =

System.out.println ("Enter number of subjects");

int n = sc.nextInt();

credits = new int [n];

marks = new int [n];

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

| Date _____ Page _____ | Date _____ Page _____ |
|---|--|
| Public static void main(String[] args) { Student student = new Student(); student.accepts_marks(); student.display_marks(); System.out.println("SGPA : " + student.calculate_SGPA()); 2 3 | Enter marks for Subject 2 98 Enter marks for Subject 3 99 USN : L24 Name : Royce Marks Subject 1 : 97 Subject 2 : 99 Subject 3 : 98 |
| <u>Output</u> Enter USN : L24 | credits Subject 1 : 4 Subject 2 : 3 Subject 3 : 2 |
| Enter Name Royce | SGPA : 10.0 |
| Enter number of subject 3 | |
| Enter credit for Subject 1 4 | |
| Enter marks for Subject 1 97 | |

```

Date / / 
Page

System.out.println("Enter credits of subjects +(i+1)+:");
credits[i] = Sc.nextInt();
}

System.out.println("Enter marks for Subject " +(i+1) + ":" );
marks[i] = Sc.nextInt();
}

Public void Details() {
System.out.print(" USN :" + USN);
System.out.print(" Name :" + name);
System.out.print(" Marks :" );
}

For ( int i=0 ; i< marks.length ; i++ ) {
System.out.println("Subject " +(i+1) + ":" + marks[i]);
}

System.out.println(" credits :" );
}

For ( int i=0 ; j < credits.length ; j++ ) {
System.out.print(" Subject " +(i+1) + ":" + credits[i]);
}

Public double calculate SGPA () {
double total Grade = 0;
int total credit = 0;
for ( int i=0 ; i< credits.length ; i++ ) {
}
}

Total Grade += get grade (marks[i]) * credits[i];
total credit += credit[i];
}

return total grade / total credit;
}

Private double get grade ( int marks ) {
if ( marks >= 90 ) {
return 10;
}
else if ( marks >= 80 ) {
return 9;
}
else if ( marks >= 70 ) {
return 8;
}
else if ( marks >= 60 ) {
return 7;
}
else if ( marks >= 50 ) {
return 6;
}
else {
return 0;
}
}

```

LABORATORY PROGRAM - 3

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

```
import java.util.Scanner;

class Book
{
    String name;
    String author;
    double price;
    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 setName(String name)
    {
        this.name = name;
    }

    public String getName()
    {
        return name;
    }

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

    public String getAuthor()
    {
        return author;
    }

    public void setPrice(double price)
    {
```

```

        this.price = price;
    }

    public double getPrice()
    {
        return price;
    }

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

    public int getNumPages()
    {
        return num_pages;
    }

    public String toString()
    {
        return "Name: " + name + "\nAuthor: " + author + "\nPrice: " + price + "\nNumber of Pages: " +
num_pages;
    }

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

```

OUTPUT

```
D:\NotePad++\Java>java Book
Enter the number of books:
2
Enter details for book 1:
Enter name:
Rog
Enter author:
Zephyrus
Enter price:
560
Enter number of pages:
1536
Enter details for book 2:
Enter name:
Power
Enter author:
Murphy
Enter price:
399
Enter number of pages:
256
Details of book 1:
Name: Rog
Author: Zephyrus
Price: 560.0
Number of Pages: 1536
Details of book 2:
Name: Power
Author: Murphy
Price: 399.0
Number of Pages: 256
```

Program 3:

- Q. Create a class Book which contains four members name, author, price & numPages. include a constructor to set the values for the members. include methods to set and get the details of the object. include a toString() method that could display the complete details of the book. Develop a Java program to create a books object.

Sol:
import java.util.Scanner;
class Book {
 String name;
 String author;
 Double price;
 int numPages;

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

void setDetails (String name, String author
Double price, int numPages)
{
 this.name = name;
 this.author = author;
 this.price = price;
 this.numPages = numPages;

void getDetails()

{
 String s = "Book :" + name + " by author :" + author +
 " with pages :" + numPages + " is of price :" + price;
}

System.out.println(s);

}

}

Public String toString()

{
 String s = "Book :" + name + " by author :" + author +
 " with pages :" + numPages + " is of price :" + price;
}

return s;

}

}

Public class Main

Public static void main (String [] args)

{

Scanner s = new Scanner (System.in);
System.out.print ("Enter the no. of book to create");
int n = s.nextInt();

Book [] books = new Book (n);

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

{

System.out.print ("Enter details for Book " + i + " ");

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

String name = s.nextLine();

System.out.print (" Author : ");

Date: / /
 Page: _____

String author = Scanner.nextLine();
 System.out.print("Price : ");
 Double price = scanner.nextDouble();
 System.out.print("No. of Pages : ");
 int numPages = scanner.nextInt();
 book[i] = new Book(name, author, price, numPages);

}
 System.out.println("Enter details of books :");
 for (int i = 0; i < n; i++) {
 System.out.println("In book " + (i + 1) + "\n" + book[i]);
 . . . to String(i));
 }

S. close();

}
 }
Output:-
 constructor values:
 Name: Mehta
 Author: neil
 Price - 56
 No. of Pages = 154

Date: / /
 Page: _____

Enter the details of book 1
 abc

Enter the name of the book
 abc

Enter the name of the author
 xyz

Enter the Price
 1000

Enter the No. of Pages
 50

Details of the book(1)

Name = abc

author = xyz

Price = 1000

Pages = 50

Algorithm:-

- (1) Start
- (2) initialize class book with name, author, price,
- (3) Read no. books.
- (4) get details of each other to of each book using for loop.
- (5) for ($i=0$; $i < n$; $i++$)
- (6) End

LABORATORY PROGRAM - 4

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

```
import java.util.Scanner;

abstract class Shape
{
    abstract void printArea();
    int length,breadth;
}

class Rectangle extends Shape
{
    Rectangle(int l,int b)
    {
        length=l;
        breadth=b;
    }

    void printArea()
    {
        int area=length*breadth;
        System.out.println("Area of Rectangle is "+area);
    }
}

class Triangle extends Shape
{
    Triangle(int l,int b)
    {
        length=l;
        breadth=b;
    }

    void printArea()
    {
        double area=0.5*length*breadth;
        System.out.println("Area of Triangle is "+area);
    }
}

class Circle extends Shape
```

```

{
    Circle(int r)
    {
        length=r;
    }

    void printArea()
    {
        double area=3.14*length*length;
        System.out.println("Area of Circle is "+area);
    }
}

class Display
{
    public static void main(String sx[])
    {
        Scanner s1=new Scanner(System.in);
        Rectangle r1=new Rectangle(0,0);
        System.out.println("Enter the Length and Breadth of Rectangle, to get it's Area: ");
        int l=s1.nextInt();
        int b=s1.nextInt();
        r1=new Rectangle(l,b);
        r1.printArea();
        Triangle t1=new Triangle(0,0);
        System.out.println("Enter the base and height of triangle, to get it's Area: ");
        int bs=s1.nextInt();
        int h=s1.nextInt();
        t1=new Triangle(bs,h);
        t1.printArea();
        Circle c1=new Circle(0);
        System.out.println("Enter the Radius of Circle, to get it's Area: ");
        int r=s1.nextInt();
        c1=new Circle(r);
        c1.printArea();
    }
}

```

OUTPUT

```
D:\NotePad++\Java>javac Shape.java

D:\NotePad++\Java>java Display
Enter the Length and Breadth of Rectangle, to get it's Area:
2
3
Area of Rectangle is 6
Enter the base and height of triangle, to get it's Area:
2
4
Area of Triangle is 4.0
Enter the Radius of Circle, to get it's Area:
2
Area of Circle is 12.56
```

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 and circle such that each one shape. Each one of the classes contain only the method printArea() that prints the area of the given shape.

~~class~~ import java.util.Scanner;

abstract class Shape {

int dimension1;

int dimension2;

public abstract void printArea();

}

class Rectangle (int length, int width) {

this.dimension1 = length;

this.dimension2 = width;

}

public void printArea () {

int area = dimension1 * dimension2;

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

}

class Triangle extends Shape

{

public Triangle (int base, int height)

{

this.dimension1 = base;

this.dimension2 = height;

}

public void printArea ()

{

int area = dimension1 * dimension2 ;

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

}

class Circle extends Shape

{

public Circle (int radius)

{

this.dimension1 = radius;

}

public void printArea ()

int area = Math.PI * dimension1 * dimension1;

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

}

}

public class Main

public static void main (String [] args)

{

Scanner scanner = new Scanner (System.in);

System.out.println ("choose a shape to calculate area
1. Rectangle 2. Triangle 3. Circle
4. Exit");

int run = 1;

while (run == 1)

{

System.out.println ("Enter choice : ");

| Date _____ | Page _____ |
|---|--|
| <pre> while (num == 1) { System.out.print("Enter choice"); int choice = Scanner.nextInt(); switch (choice) { case 1: System.out.print("Enter the length and breath :"); double l = Scanner.nextDouble(); double b = Scanner.nextDouble(); r.getData(l,b); r.printArea(); break; case 2: System.out.print("Enter base of triangle"); int base = Scanner.nextInt(); System.out.print("Enter height of triangle:"); int height = Scanner.nextInt(); ist height = Scanner.nextInt(); Triangle t = new Triangle (base, height); d = printArea(); break; } } </pre> | <pre> Case 3: System.out.print("Enter radius of circle:"); int radius = Scanner.nextInt(); Circle c = new Circle (radius); c.printArea(); break; Case 4: num = 0 break; Default: System.out.print("Invalid choice"); } Scanner.close(); </pre> |
| | 3 |
| | 3. |
| | <u>OUTPUT :-</u> |
| | <ol style="list-style-type: none"> 1. Area of rectangle 2. Area of Triangle 3. Area of circle <p>Enter your choice 3 Enter radius 5. Area of circle = 78.55 8/10/14</p> |

LABORATORY 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.

```
import java.util.Scanner;

class Account
{
    String customerName;
    long accountNumber;
    String accountType;
    double balance;

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

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

    public void displayBalance()
    {
        System.out.println("Account Number: " + accountNumber);
    }
}
```

```

        System.out.println("Customer Name: " + customerName);
        System.out.println("Account Type: " + accountType);
        System.out.println("Balance: " + balance);
    }
}

class SavAcct extends Account
{
    public SavAcct(String customerName, long accountNumber, double balance)
    {
        super(customerName, accountNumber, "Savings", balance);
    }

    public void computeAndDepositInterest(double rate)
    {
        double interest = balance * rate / 100;
        balance += interest;
        System.out.println("Interest computed and deposited. Updated balance: " + balance);
    }

    public void withdraw(double amount)
    {
        if (amount <= balance)
        {
            balance -= amount;
            System.out.println("Withdrawal successful. Updated balance: " + balance);
        }
        else
        {
            System.out.println("Insufficient funds. Withdrawal failed.");
        }
    }
}

class CurrAcct extends Account
{
    double minimumBalance;
    double serviceCharge;

    public CurrAcct(String customerName, long accountNumber, double balance, double minimumBalance, double serviceCharge)
    {
        super(customerName, accountNumber, "Current", balance);
        this.minimumBalance = minimumBalance;
        this.serviceCharge = serviceCharge;
    }

    private void checkMinimumBalance()

```

```

{
    if (balance < minimumBalance)
    {
        balance -= serviceCharge;
        System.out.println("Minimum balance not maintained. Service charge imposed. Updated
balance: " + balance);
    }
}

public void withdraw(double amount)
{
    if (amount <= balance)
    {
        balance -= amount;
        System.out.println("Withdrawal successful. Updated balance: " + balance);
        checkMinimumBalance();
    }
    else
    {
        System.out.println("Insufficient funds. Withdrawal failed.");
    }
}
}

public class Bank
{
    public static void main(String[] args)
    {
        Scanner s1 = new Scanner(System.in);

        System.out.print("Enter customer name for Savings Account: ");
        String SCN = s1.nextLine();
        System.out.print("Enter account number for Savings Account: ");
        long SAN = s1.nextLong();
        System.out.print("Enter initial balance for Savings Account: ");
        double SIB = s1.nextDouble();
        SavAcct SA = new SavAcct(SCN, SAN, SIB);

        System.out.print("Enter customer name for Current Account: ");
        String CCN = s1.next();
        System.out.print("Enter account number for Current Account: ");
        long CAN = s1.nextLong();
        System.out.print("Enter initial balance for Current Account: ");
        double CIB = s1.nextDouble();
        System.out.print("Enter minimum balance for Current Account: ");
        double MB = s1.nextDouble();
        System.out.print("Enter service charge for Current Account: ");
        double SC = s1.nextDouble();
    }
}

```

```
CurrAcct CA = new CurrAcct(CCN, CAN, CIB, MB, SC);

System.out.print("Enter deposit amount for Savings Account: ");
double SDA = s1.nextDouble();
SA.deposit(SDA);

System.out.print("Enter interest rate for Savings Account: ");
double SIR = s1.nextDouble();
SA.computeAndDepositInterest(SIR);

System.out.print("Enter withdrawal amount for Savings Account: ");
double SWA = s1.nextDouble();
SA.withdraw(SWA);

System.out.print("Enter deposit amount for Current Account: ");
double CDA = s1.nextDouble();
CA.deposit(CDA);

System.out.print("Enter withdrawal amount for Current Account: ");
double CWA = s1.nextDouble();
CA.withdraw(CWA);

System.out.println("\nFinal Balances:");
System.out.println("Savings Account:");
SA.displayBalance();

System.out.println("\nCurrent Account:");
CA.displayBalance();

}
```

OUTPUT

```
D:\NotePad++\Java>javac Bank.java

D:\NotePad++\Java>java Bank
Enter customer name for Savings Account: Ram
Enter account number for Savings Account: 2324
Enter initial balance for Savings Account: 5000
Enter customer name for Current Account: Ram
Enter account number for Current Account: 2324
Enter initial balance for Current Account: 6000
Enter minimum balance for Current Account: 1000
Enter service charge for Current Account: 100
Enter deposit amount for Savings Account: 2000
Deposit successful. Updated balance: 7000.0
Enter interest rate for Savings Account: 2
Interest computed and deposited. Updated balance: 7140.0
Enter withdrawal amount for Savings Account: 500
Withdrawal successful. Updated balance: 6640.0
Enter deposit amount for Current Account: 1000
Deposit successful. Updated balance: 7000.0
Enter withdrawal amount for Current Account: 750
Withdrawal successful. Updated balance: 6250.0

Final Balances:
Savings Account:
Account Number: 2324
Customer Name: Ram
Account Type: Savings
Balance: 6640.0

Current Account:
Account Number: 2324
Customer Name: Ram
Account Type: Current
Balance: 6250.0
```

LAD Program -5

Date / /
Page

Develop a Java Program to create a class Bank that maintains two kinds of account for its customer one called Saving & Saving account and the other Current account. The Saving account provides compound interest and withdrawal facilities but no cheque book facility.

- (a) Accept deposit from customer and update the balance
- (b) Display the balance
- (c) compute and deposit interest
- (d) Permit withdrawal and update balance

→ import java.util.Scanner;
class Account

{
String customerName;
long accountNumber;
double balance;

Public Account (String customerName, long account
Number, String accountType,
double balance)

g

class over Acct extends Account
 {
 double minimum Balance;
 double Service charge;
 public construct (String customer name, long account number, double balance,
 double minimum balance,
 double service charge)
 {
 super (customer name, account Number, "current"
 , balance);
 this minimum Balance = minimum Balance;
 this Service charge = Service charge;
 }
 private void check (Minimum Balance)
 {
 if (balance < minimum balance)
 {
 balance - = Service charge;
 System.out.println ("Minimum balance Not
 maintained. Service charge
 imposed. updated balance :"
 + balance);
 }
 }
 public void withdraw (double amount)
 {
 if (amount < balance)
 {
 balance - = amount;
 System.out.println ("withdrawal successful. updated
 balance :" + balance);
 check minimum balance();
 }
 else
 {
 System.out.println ("Insufficient funds. withdrawal
 failed");
 }
 }
 public class Bank
 {
 public static void main (String [] args)
 {
 Scanner s1 = new Scanner (System.in);
 System.out.println ("Enter customer name for
 saving account :");
 String SCN = s1.nextLine();
 }

System.out.print ("Enter account number for
 current account"),
 long CAN = s1.nextInt();
 System.out.print ("Enter initial balance for current
 account");
 Double CIB = s1.nextDouble();
 System.out.print ("Enter minimum balance for
 current account");
 Double MB = s1.nextDouble();
 System.out.print ("Enter Service charge for current
 account");
 Double SC = s1.nextDouble();
 curr Acct CA = new curr Acct (CCN, CAN, CIB, MB, SC);
 System.out.print ("Enter interest rate for Saving
 account");
 Double SIR = s1.nextDouble();
 SA deposit (SIR);
 System.out.print ("Enter interest rate for Saving account");
 Double SIR = s1.nextDouble();
 SA. compute and Deposit interest (SIR);

System.out.print ("Enter withdrawal amount for
 saving account");
 Double SWA = s1.nextDouble();
 SA. withdraw (SWA);
 System.out.print ("Enter deposit amount of for
 current account");
 Double CWA = s1.nextDouble();
 CA. withdraw (CWA);
 System.out.print ("Enter deposit amount for
 current account");
 Double CDW = s1.nextDouble();
 CA. deposit (CDW);
 System.out.print ("(" + current account + ");
 CA. display Balance();

Date _____
Page _____

OUTPUT

Final Balance :-
Saving account :-

Account Number :- 2324
Customer name : Raja.

Account Type : Saving:-
Balance : 6690.00

Current Account :-
Account Number : 2324
Customer name = Raja.
Account type = current
Balance : 6250.00

LABORATORY 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 which is a derived class of Student and 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.

```
package CIE;

public class Internals extends Student
{
    public int[] internalMarks;

    public Internals(String usn, String name, int sem, int[] internalMarks)
    {
        super(usn, name, sem);
        this.internalMarks = internalMarks;
    }
}

package CIE;

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

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

package SEE;
import CIE.Student;

public class External extends Student
{
    public int[] seeMarks;
```

```

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

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

public class CalculateFinalMarks
{
    public static void main(String[] args)
    {
        Scanner s1 = new Scanner(System.in);

        System.out.println("Enter the number of students:");
        int n = s1.nextInt();

        Internals[] CS = new Internals[n];
        for (int i = 0; i < n; i++)
        {
            System.out.println("Enter details for CIE student " + (i + 1));
            System.out.print("USN: ");
            String usn = s1.next();
            System.out.print("Name: ");
            String name = s1.next();
            System.out.print("Semester: ");
            int sem = s1.nextInt();
            System.out.println("Enter internal marks for 5 courses:");
            int[] internalMarks = new int[5];
            for (int j = 0; j < 5; j++)
            {
                System.out.print("Course " + (j + 1) + ": ");
                internalMarks[j] = s1.nextInt();
            }
            CS[i] = new Internals(usn, name, sem, internalMarks);
        }

        External[] SS = new External[n];
        for (int i = 0; i < n; i++)
        {
            System.out.println("Enter details for SEE student " + (i + 1));
            System.out.print("USN: ");

```

```

String usn = s1.next();
System.out.print("Name: ");
String name = s1.next();
System.out.print("Semester: ");
int sem = s1.nextInt();
System.out.println("Enter SEE marks for 5 courses:");
int[] seeMarks = new int[5];
for (int j = 0; j < 5; j++)
{
    System.out.print("Course " + (j + 1) + ": ");
    seeMarks[j] = s1.nextInt();
}

SS[i] = new External(usn, name, sem, seeMarks);
}

int[][] finalMarks = new int[n][5];
for (int i = 0; i < n; i++)
{
    for (int j = 0; j < 5; j++)
    {
        finalMarks[i][j] = CS[i].internalMarks[j] + SS[i].seeMarks[j];
    }
}

System.out.println("\nFinal Marks:");
for (int i = 0; i < n; i++)
{
    System.out.print("USN: " + CS[i].usn +", Name: " + CS[i].name +", Semester: " + CS[i].sem
+", Final Marks: ");
    for (int j = 0; j < 5; j++)
    {
        System.out.print(finalMarks[i][j] + " ");
    }
    System.out.println();
}
}
}

```

OUTPUT

```
D:\NotePad++\Java\Packages>javac CalculateFinalMarks.java

D:\NotePad++\Java\Packages>java CalculateFinalMarks
Enter the number of students:
1
Enter details for CIE student 1
USN: 1
Name: Ram
Semester: 3
Enter internal marks for 5 courses:
Course 1: 47
Course 2: 48
Course 3: 49
Course 4: 50
Course 5: 49
Enter details for SEE student 1
USN: 1
Name: Ram
Semester: 3
Enter SEE marks for 5 courses:
Course 1: 48
Course 2: 49
Course 3: 47
Course 4: 50
Course 5: 50

Final Marks:
USN: 1, Name: Ram, Semester: 3, Final Marks: 95 97 96 100 99
```

LAB Session-6

create a package CSE which has base class - Student and internal. The class Student has members like name, rollno, sem. The class internal which is a derived class of Student and has a array that stores the internal marks scored in five courses of the current Student of the Student.

→ Package CSE;

Public class internal extends Student
2.

Public int[] internal marks;

Public internal (String name, String rollno, int sem
internal marks)

Super (name, rollno, sem);
this internal marks = internal marks;

3

Package CSE;

Public class Student

4.

Public String name;

Public String rollno;

Public int sem;

System.out.println("causa + (j++) + "1");
internal_marks[i] = si.nextInt();

?
cc(i) = new internal(i, name, len, internal_marks);

?
ExternalITS = new External(i);
for (int id = 0; id < i; id++)

System.out.println("Enter details for set Student");
(id++));

System.out.print(" ");

String usn = si.nextLine();
String name = si.nextLine();
String len = si.nextLine();

System.out.print("Board");
int sum = si.nextInt();

System.out.println("Enter SF marks for " + name + " ");

int[] fee_marks = new int[sum];
for (int j = 0; j < sum; j++)

?

higher_out.print("causa");
(j++);

fee_marks[i] = si.nextInt();

?
SF_marks[i] =

SFITS = new External(i);

?

int [IT] final_marks = new

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

?

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

?

Final_marks[ITS] = SFITS;

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

?

System.out.println("Final");

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

?

System.out.println("USN: " + SFITS);

else {

 System.out.println("Final marks");

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

?

 System.out.println("Final marks");

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

?

 System.out.println("Final marks");

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

?

 System.out.println("Final marks");

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

?

 System.out.println("Final marks");

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

?

 System.out.println("Final marks");

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

?

 System.out.println("Final marks");

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

?

 System.out.println("Final marks");

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

?

 System.out.println("Final marks");

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

?

 System.out.println("Final marks");

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

?

 System.out.println("Final marks");

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

?

 System.out.println("Final marks");

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

?

 System.out.println("Final marks");

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

?

 System.out.println("Final marks");

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

?

 System.out.println("Final marks");

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

?

 System.out.println("Final marks");

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

?

 System.out.println("Final marks");

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

?

 System.out.println("Final marks");

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

?

 System.out.println("Final marks");

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

?

 System.out.println("Final marks");

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

?

 System.out.println("Final marks");

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

?

 System.out.println("Final marks");

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

?

 System.out.println("Final marks");

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

?

 System.out.println("Final marks");

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

?

 System.out.println("Final marks");

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

?

 System.out.println("Final marks");

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

?

 System.out.println("Final marks");

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

?

 System.out.println("Final marks");

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

?

 System.out.println("Final marks");

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

?

 System.out.println("Final marks");

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

?

 System.out.println("Final marks");

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

?

 System.out.println("Final marks");

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

?

 System.out.println("Final marks");

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

?

 System.out.println("Final marks");

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

?

 System.out.println("Final marks");

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

?

 System.out.println("Final marks");

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

Palle Model (Many vs. Many vs. Many)

$$\begin{aligned} \text{Palle} &= 100 \\ \text{Palle} &= 100 \\ \text{Palle} &= 100 \end{aligned}$$

?

Package SSE:
infot
CSE Student:

Palle still extends outside Student

Palle is not far enough

Palle Interval (Many vs. Many one, either, will be far)

?

(label very, now, few);
the far ends + far ends

?

infot, jne, will become
infot, jne
infot, jne

Palle still extends outside

Palle still near end (Many (S op))

?

lepton and had ("extra distance for CSE")

infot = S. next (S)

Interval (CSE) = many intervals (CSE)
(or (infot 0, 100, 100))

?

lepton and had ("extra distance for CSE")

+ (S op))

lepton and had ("many")

many many = S. next (S)

lepton and had ("extra distance for CSE")

infot = S. next (S)

?

lepton and had ("extra distance for CSE")

infot = S. next (S)

lepton and had ("extra distance for CSE")

+ (S op))

```
For(i=0 ; i<5 ; i++)
```

```
{  
    System.out.print("Final marks : " + arr[i]);  
}
```

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

```
}
```

```
}
```

OUTPUT

Enter the details number of Student 1 :-

Enter details for CIE student 1

USN 1

Name : Ram

Semester : 3

Enter internal marks for 5 course

Course 1 : 47

Course 2 : 48

Course 3 : 49

Course 4 : 50

Course 5 : 49

Final marks

USN 1

Name Ram, Semester 3.

Final marks:

95 97 96 100 99

LABORATORY PROGRAM - 7

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 cases both father and son’s age and throws an exception if son’s age is >=father’s age.

```
import java.util.Scanner;

class WrongAge extends Exception
{
    public WrongAge()
    {
        super("Invalid age! Age cannot be negative nor zero.");
    }

    public WrongAge(String message)
    {
        super(message);
    }
}

class Father
{
    private int age;

    public Father(int age) throws WrongAge
    {
        if (age <= 0)
        {
            throw new WrongAge();
        }
        this.age = age;
    }

    public int getAge()
    {
        return age;
    }
}

class Son extends Father
{
    private int sonAge;
```

```

public Son(int fatherAge, int sonAge) throws WrongAge
{
    super(fatherAge);

    if (sonAge >= fatherAge)
    {
        throw new WrongAge("Son's age should be less than Father's age.");
    }

    this.sonAge = sonAge;
}

public int getSonAge()
{
    return sonAge;
}

public class InheritanceException
{
    public static void main(String[] args)
    {
        try
        {
            Scanner s1 = new Scanner(System.in);

            System.out.print("Enter Father's age: ");
            int fatherAge = s1.nextInt();
            Father f = new Father(fatherAge);

            System.out.print("Enter Son's age: ");
            int sonAge = s1.nextInt();
            Son s = new Son(f.getAge(), sonAge);

            System.out.println("Father's age: " + f.getAge());
            System.out.println("Son's age: " + s.getSonAge());

        } catch (WrongAge e)
        {
            System.out.println("Exception: " + e);
        } catch (Exception e)
        {
            System.out.println("Exception: Invalid input. Please enter valid integer values.");
        }
    }
}

```

OUTPUT

```
D:\NotePad++\Java>javac InheritanceException.java

D:\NotePad++\Java>java InheritanceException
Enter Father's age: 46
Enter Son's age: 56
Exception: WrongAge: Son's age should be less than Father's age.

D:\NotePad++\Java>java InheritanceException
Enter Father's age: 56
Enter Son's age: 24
Father's age: 56
Son's age: 24
```

Program - F

f. write a java program that demonstrate handling of exceptions in inheritance tree. create a base class "Father" and derived class d called "Son" which extends the base class. In Father class implement a constructor which takes the age and throws the exception wrong Age() when the input age < 0. In Son class, implement a constructor that cases both Father and Son age and throws an exception if Son age and throws an exception if Son age is \neq Father age.

Sol: class wrongAge extends Exception {

 Public wrong Age (String message) {

 Super (message);

 }

 }

class Father {

 Private int age;

 Public Father (int age) throws wrong Age {

 if (age < 0) {

 throw new wrong Age ("Age cannot be negative");

 }

 this.age = age;

 }

 Public int getAge() {

 return age;

 }

 }

OUTUT Father age
 Son age = 20 Date / /
 Page

class Son extends Father {
 public int Son Age;
 }

Public Son (int Father Age, int Son Age) throws wrong Age
 Super (Father Age);
 if (Son Age >= Father Age) {
 throw new wrong Age ("Son age cannot be greater than
 or equal to Father age");
 }
 this. Son Age = Son Age;

g
 Public int get Son Age () {
 return Son Age;
 }
 g

Public class Exception Inheritance Demo {
 Public static void main (String[] args) {
 try {
 Father Father = new Father (25);
 Son Son = new Son (45, 40);
 System.out.println ("Father's age :" + Father.get Age());
 System.out.println ("Son's age :" + Son.get SonAge());
 catch (wrong Age e) {
 System.out.println ("Exception caught :" + e.get Message());
 }
 }

LABORATORY PROGRAM - 8

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

```
class BMSThread implements Runnable
{
    public void run()
    {
        while (true)
        {
            try
            {
                System.out.println("BMS College of Engineering");
                Thread.sleep(10000);
            }
            catch (InterruptedException ie)
            {
                System.out.println("BMSThread is Interrupted");
            }
        }
    }
}

class CSEThread implements Runnable
{
    public void run()
    {
        while (true)
        {
            try
            {
                System.out.println("CSE");
                Thread.sleep(2000);
            }
            catch (InterruptedException ie)
            {
                System.out.println("CSEThread is Interrupted");
            }
        }
    }
}

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

```
{  
    Thread bms = new Thread(new BMSThread());  
    Thread cse = new Thread(new CSEThread());  
    bms.start();  
    cse.start();  
}  
}
```

OUTPUT

```
D:\NotePad++\Java>javac Display.java
```

```
D:\NotePad++\Java>java Display
```

```
BMS College of Engineering
```

```
CSE
```

```
BMS College of Engineering
```

```
CSE
```

```
BMS College of Engineering
```

```
CSE
```

```
CSE
```

```
|
```

Program 8:

Date / /
Page

- f. 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.

soln
class displayMessage extends Thread {
String message;
int interval;

Public Display Messages (String message , int interval) {

this. message = message;

this. interval = interval;

}

Public void run () {

while (true) {

try {

System.out.println (message);

Thread.sleep (interval * 1000);

}

catch (InterruptedException e) {

e.printStackTrace();

}

}

}

Public class Thread Demo {

Public static void main (String [] args) {

Display Message Thread 1 = new display Messages
("BMS college of Engineering")

Display message Thread 2 = New display message ("CSE")

Thread 1. Start ()

Thread 2. Start ()

3

3.

Output

BMS college of Engineering.

CSE

CSE

CSE

CSE

CSE.

BMS college of Engineering.

CSE

CSE

CSE

CSE

CSE.

81 10/10

LABORATORY PROGRAM - 9

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

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

public class DivisionMain1 extends Frame implements ActionListener
{
    TextField num1,num2;
    Button dResult;
    Label outResult;
    String out="";
    double resultNum;
    int flag=0;

    public DivisionMain1()
    {
        setLayout(new FlowLayout());

        dResult = new Button("RESULT");
        Label number1 = new Label("Number 1:",Label.RIGHT);
        Label number2 = new Label("Number 2:",Label.RIGHT);
        num1=new TextField(5);
        num2=new TextField(5);
        outResult = new Label("Result:",Label.RIGHT);

        add(number1);
        add(num1);
        add(number2);
        add(num2);
        add(dResult);
        add(outResult);

        num1.addActionListener(this);
        num2.addActionListener(this);
        dResult.addActionListener(this);
        addWindowListener(new WindowAdapter()
        {
            public void windowClosing(WindowEvent we)
            {
                System.exit(0);
            }
        });
    }
}
```

```

        }
    });

}

public void actionPerformed(ActionEvent ae)
{
    int n1,n2;
    try
    {
        if (ae.getSource() == dResult)
        {
            n1=Integer.parseInt(num1.getText());
            n2=Integer.parseInt(num2.getText());

            /*if(n2==0)
                throw new ArithmeticException();*/
            out=n1+" "+n2+" ";
            resultNum=n1/n2;
            out+=String.valueOf(resultNum);
            repaint();
        }
    }
    catch(NumberFormatException e1)
    {
        flag=1;
        out="Number Format Exception! "+e1;
        repaint();
    }
    catch(ArithmeticException e2)
    {
        flag=1;
        out="Divide by 0 Exception! "+e2;
        repaint();
    }
}

public void paint(Graphics g)
{
    if(flag==0)
        g.drawString(out,outResult.getX()+outResult.getWidth(),outResult.getY()+outResult.getHeight()-8);
    else
        g.drawString(out,100,200);
    flag=0;
}

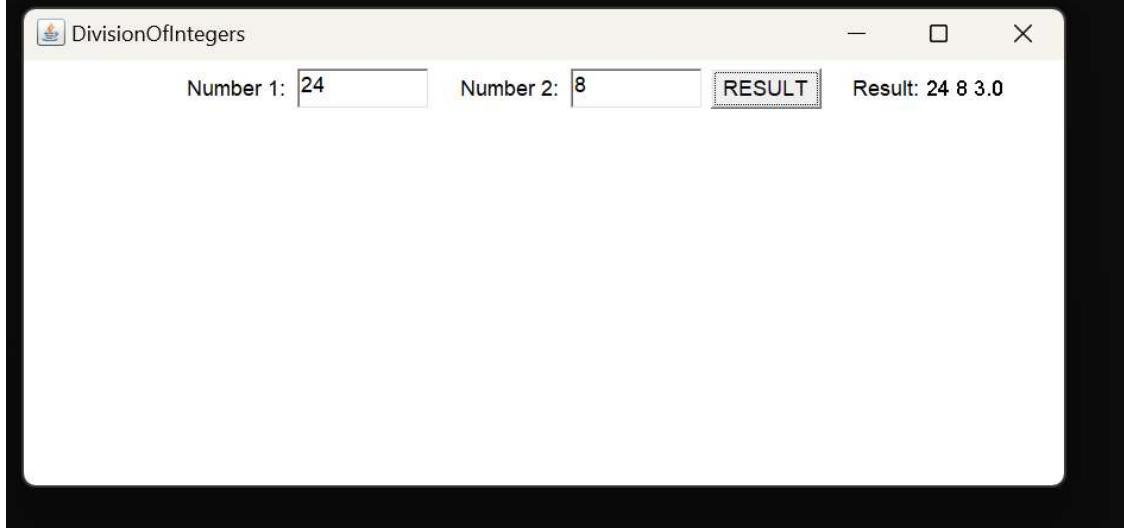
```

```
public static void main(String[] args)
{
    DivisionMain1 dm=new DivisionMain1();
    dm.setSize(new Dimension(800,400));
    dm.setTitle("DivisionOfIntegers");
    dm.setVisible(true);
}
}
```

OUTPUT

```
D:\NotePad++\Java>javac DivisionMain1.java
```

```
D:\NotePad++\Java>java DivisionMain1
```



23/02/24

Date 23/1
Page

Program - 9.

- Q. 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 and Num2 are not integer, the Program could throw a Number Format exception. If Num2 were zero, the Program could throw an arithmetic exception. Display the exception in a message dialog box.

Soln

CODE :-

```
import java.awt.*;  
import java.awt.event.*;  
  
class Swing Demo extends JFrame  
{  
    JButton Divide = new JButton("Divide");  
    JTextField Num1 = new JTextField("Enter the Dividend");  
    JTextField Num2 = new JTextField("Enter the Divisor");  
    JTextField Result = new JTextField("Result");  
  
    public void actionPerformed(ActionEvent e)  
    {  
        if (e.getSource() == Divide)  
        {  
            int n1 = Integer.parseInt(Num1.getText());  
            int n2 = Integer.parseInt(Num2.getText());  
            Result.setText(Integer.toString(n1 / n2));  
        }  
    }  
}  
  
public class Main  
{  
    public static void main(String args[])  
    {  
        Swing Demo obj = new Swing Demo();  
        obj.setSize(300, 300);  
        obj.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
        obj.setVisible(true);  
    }  
}
```

```

Text field aif = new JTextField(8);
Text field bif = new JTextField(8);

JButton button = new JButton("calculate");
JLabel err = new JLabel();
JLabel aif = new JLabel();
JLabel bif = new JLabel();
JLabel anslab = new JLabel();

jfrm.add(err);
jfrm.add(jlab);
jfrm.add(bif);
jfrm.add(button);
jfrm.add(aif);
jfrm.add(anslab);

ActionListener t = new ActionListener() {
    public void actionPerformed(ActionEvent evt) {
        if(jtf1.getText().length() < 1 || jtf2.getText().length() < 1) {
            err.setText("Action event from a text field!");
        } else {
            try {
                int a = Integer.parseInt(jtf1.getText());
                int b = Integer.parseInt(jtf2.getText());
                anslab.setText("\n Ans = " + a+b);
            } catch (NumberFormatException e) {
                err.setText("Enter only Integers!");
            }
        }
    }
};
button.addActionListener(t);
err.addActionListener(t);

```

```
Page  
catch (ArithmeticException e) {  
    aLab.setText("");  
    bLab.setText("");  
    ansLab.setText("");  
    err.setText("B Should be Non zero!");  
}  
}  
};  
jFrame.setVisible(true);  
}
```

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

```
SwingUtilities.invokeLater(new Runnable() {
```

```
public void run() {
```

```
new Demo();  
};  
};  
};
```

OUTPUT

```
Enter the divisor = 12
```

```
Enter the dividend = 3
```

```
A = 12 , B = 3
```

```
Ans = 4
```

LABORATORY - 10

REPORT

The given program utilize Java's AWT and Swing libraries to create GUI Applications. These programs showcase various event handling in Java.

i) ButtonDemo : Is an applet that demonstrates event handling in Java AWT. It consists of three buttons labeled "Yes", "No" and "Undecided". Clicking on each button triggers an action event, and the corresponding message is displayed on the applet.

ii) ButtonList : Is another frame-based Java Application that demonstrates event handling. And consists of three buttons similar to the ButtonDemo program. Clicking on any button updates a message indicating the button pressed.

iii) buttondrag : Is a frame-based Java app that implements a puzzle game. Here, players rearrange numbered buttons in ascending order by swapping their positions.

iv) DivisionMain : Is a frame-based Java app that allows users to input 2 numbers and calculate their division. It includes error handling for scenarios such as division by zero and invalid input formats.

