VISVESVARAYA TECHNOLOGICAL UNIVERSITY, JNANASANGAMA, BELGAUM - 590014, KARNATAKA

LABORATORY RECORD

ON

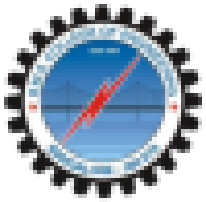
Object Oriented Java Programming

(23CS3PCOOJ) ***Submitted by***

LIKHITH M (1BM22CS135)

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

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 discriminate 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
```

18.12.23

papergrid
Date: / /

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 discriminate 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
      Confugate Imaginary Roots:");
      System.out.printf("Root 1: %.5f + %.5fi%n", real-
      Part, ImaginaryPart);
      System.out.printf("Root 2: %.5f - %.5fi%n", realPart,
      ImaginaryPart);
    }
```

Date: / /

```
    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();
```

papergrid
Date: / /

```
    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 :

```
Enter the Coefficients of Quadratic Equation:
2
3
4
The Quadratic Equation has Confugate Imaginary
Roots:
Root 1: -0.52174 + 0.92640i
Root 2: -0.52174 - 0.92640i

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
```

Date: / /

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

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
```

1-1-24

papergrid

Date: / /

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)+": ");
    }
}

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:

```
Enter USN:
1234
Enter NAME:
Royce
Enter number of subjects:
4
Enter credits for subject 1:
4
Enter marks for subject 1:
94
Enter credits for subject 2:
3
Enter marks for subject 2:
96
Enter credits for subject 3:
2
Enter marks for subject 3:
99
Enter credits for subject 4:
1
Enter marks for subject 4:
89
USN: 1234
Name: Royce
Marks:
Subject 1: 94
Subject 2: 96
Subject 3: 99
Subject 4: 89
Credits:
Subject 1: 4
Subject 2: 3
Subject 3: 2
Subject 4: 1
SGPA: 9.9
```

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
```

1.1.24

Date: / /

LABORATORY PROGRAM – 3

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;
    }
```

papergrid

Date: / /

```
    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 = new pages;
    }
    public int getNumPages()
    {
        return num_pages();
    }
    public String toString()
    {
        return "Name: " + name + "\nAuthor: " +
        author + "\nPrice: " + price + "\nNumber of
        Pages: " + num_pages;
    }
```

Date: / /

```
    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
            System.out.println("Enter price: ");
            double price = sc.nextDouble();
            System.out.println("Enter numbers 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());
        }
    }
}
```

papergrid

Date: / /

OUTPUT:

```
Enter the number of books:
2
Enter details for books:
Enter name:
Ros
Enter author:
Zephyrus
Enter price:
560
Enter number of pages:
1536
Enter details for book 2:
Enter name:
Asus
Enter author:
My
Enter price:
780
Enter number of pages:
689

Details of Book 1:
Name: Ros
Author: Zephyrus
Price: 560.0
Number of Pages: 1536
Details of Book 2:
Name: Asus
Author: My
Price: 780.0
Number of Pages: 689
```

LABORATORY PROGRAM - 4

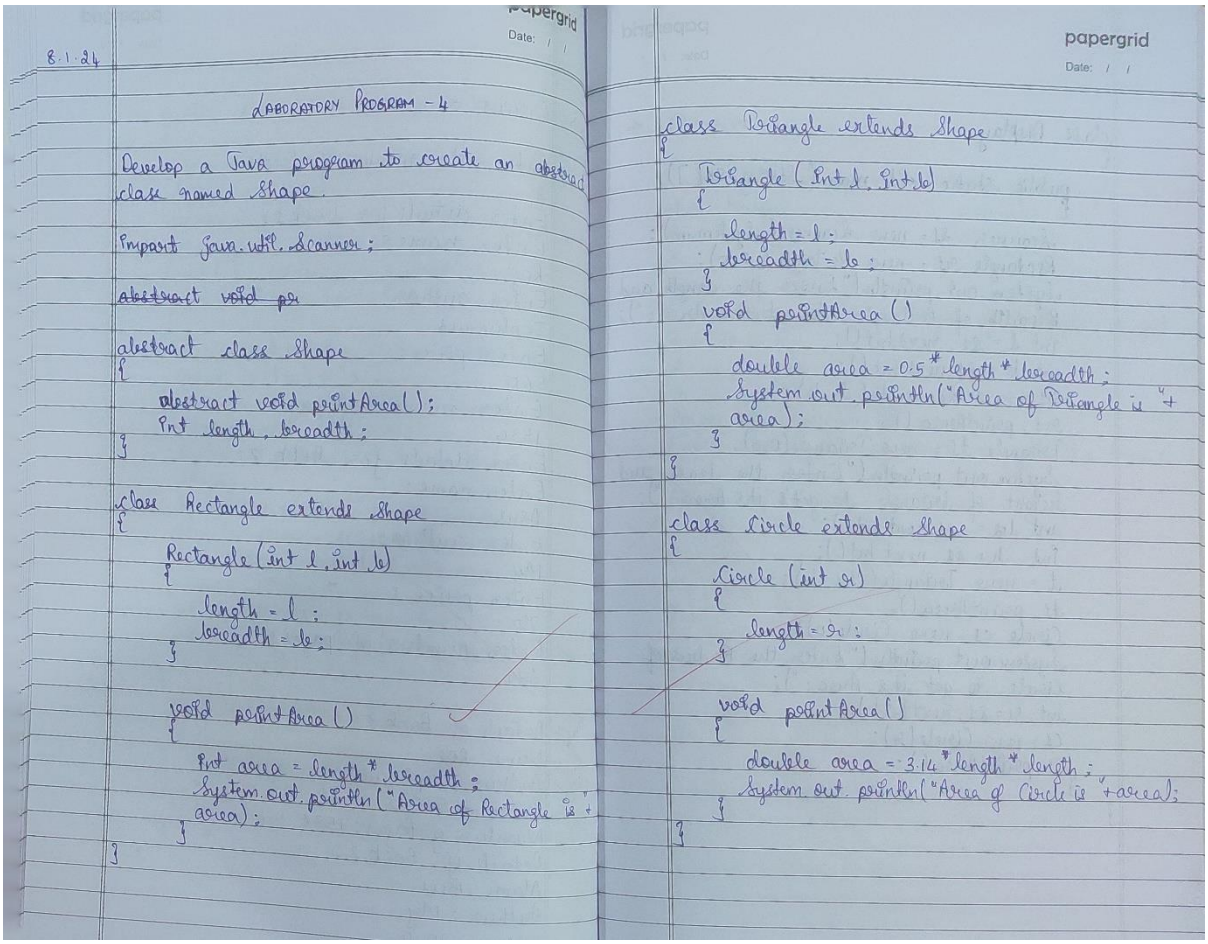
Develop a Java program to create an abstract class named Shape that contains two integers and an empty method named printArea( ). Provide three classes named Rectangle, Triangle and Circle such that each one of the classes extends the class Shape. Each one of the classes contain only the method printArea(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



8.1.24

LABORATORY PROGRAM - 4

Develop a Java program to create an abstract class named Shape.

```
import java.util.Scanner;

abstract void pa

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 sc = new Scanner (System.in);
        Rectangle rt = new Rectangle (0,0);
        System.out.println("Enter the length and
        Breadth of Rectangle, to get it's Area:");
        int l = sc.nextInt();
        int b = sc.nextInt();
        rt = new Rectangle(l, b);
        rt.printArea();
        Triangle tt = new Triangle(0,0);
        System.out.println("Enter the base and
        height of triangle, to get its Area:");
        int bas = sc.nextInt();
        int h = sc.nextInt();
        t = new Triangle(bas, h);
        tt.printArea();
        Circle ct = new Circle(0);
        System.out.println("Enter the Radius of
        Circle, to get it's Area:");
        int r = sc.nextInt();
        ct = new Circle(r);
        ct.printArea();
    }
}
```

→ OUTPUT:

```
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
```

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();

    }
}
```

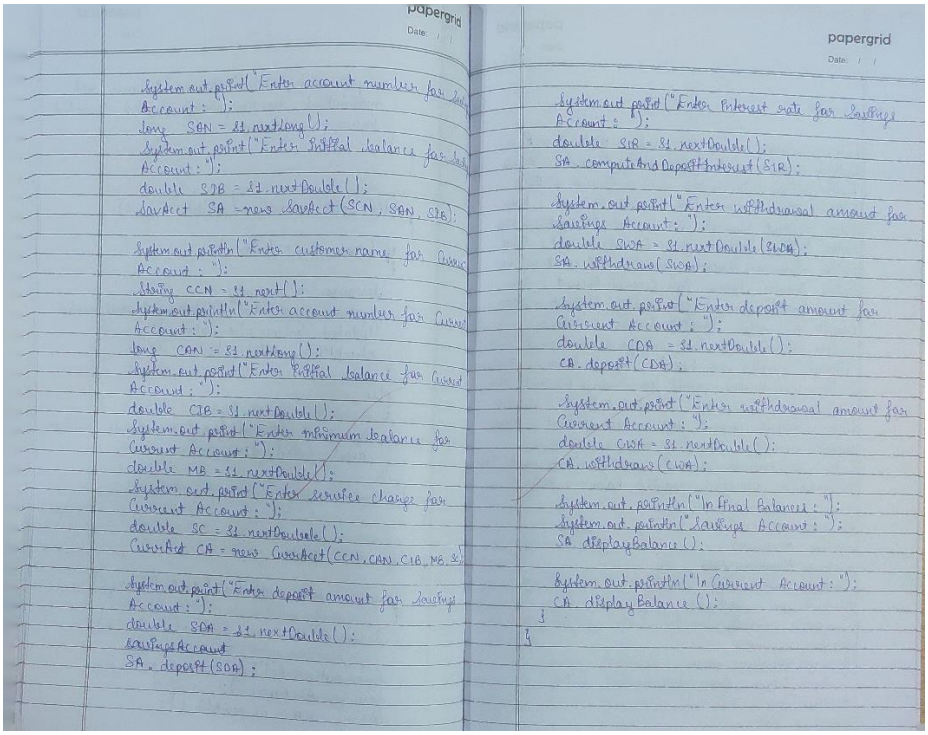
OUTPUT

```
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
```

LABORATORY - 5

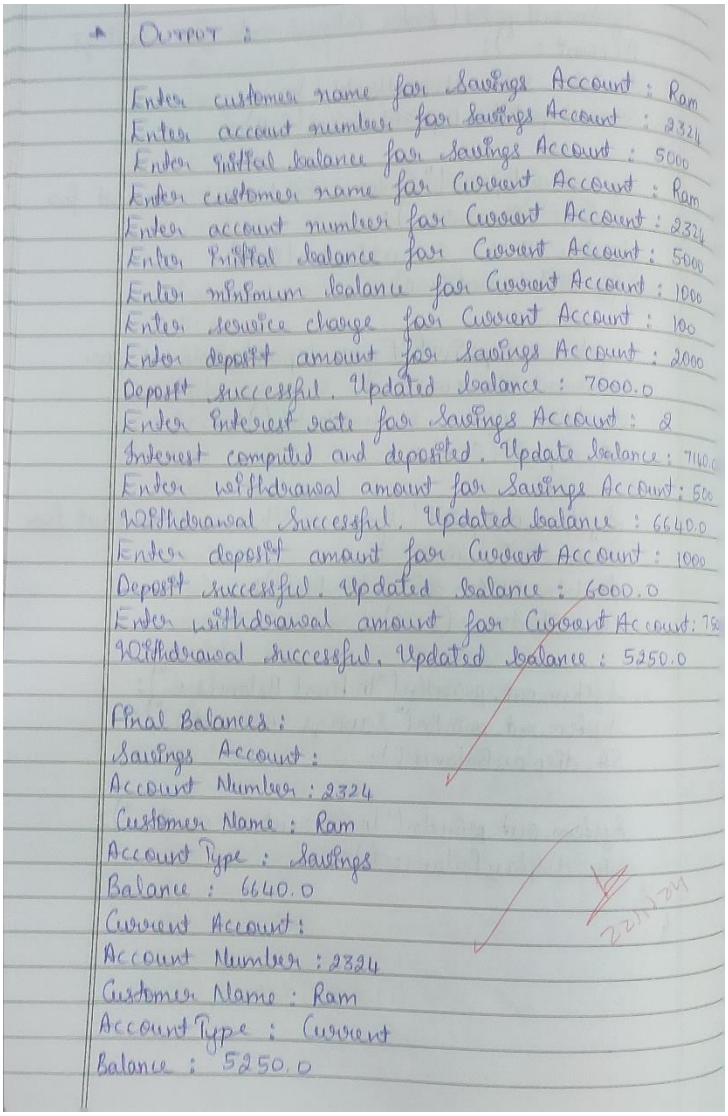
Develop a Java Program to create a class Bank that maintains two kinds of account for its customers.

```
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 sc = new Scanner(System.in);

        System.out.println("Enter customer name for Savings Account: ");
        String SCN = sc.nextLine();
```



```
System.out.println("Enter account number for Savings
Account : ");
long SAN = sc.nextLong();
System.out.println("Enter Initial balance for Savings
Account : ");
double SIB = sc.nextDouble();
SavAcct SA = new SavAcct(SCN, SAN, SIB);

System.out.println("Enter customer name for Current
Account : ");
String CCN = sc.next();
System.out.println("Enter account number for Current
Account : ");
long CAN = sc.nextLong();
System.out.println("Enter Initial balance for Current
Account : ");
double CIB = sc.nextDouble();
System.out.println("Enter minimum balance for
Current Account : ");
double MB = sc.nextDouble();
System.out.println("Enter service charge for
Current Account : ");
double SC = sc.nextDouble();
CurrAcct CA = new CurrAcct(CCN, CAN, CIB, MB, SC);

System.out.println("Enter deposit amount for Savings
Account : ");
double SDA = sc.nextDouble();
SavingsAccount
SA.deposit(SDA);
```

```
System.out.println("Enter Interest rate for Savings
Account : ");
double SIR = sc.nextDouble();
SA.computeAndDepositInterest(SIR);

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

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

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

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

System.out.println("\n Current Account : ");
CA.displayBalance();
}
}
```



```
Output :

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 : 5000
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 compited and deposited. Update 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 : 6000.0
Enter withdrawal amount for Current Account : 750
Withdrawal successful. Updated balance : 5250.0

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 : 5250.0
```

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.

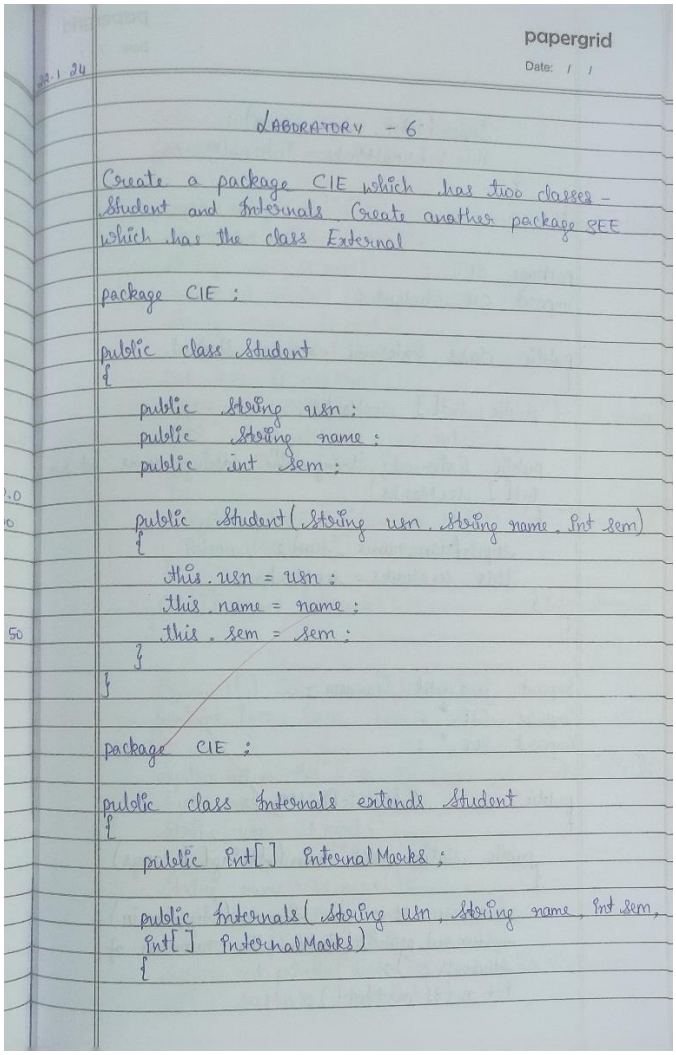
```
package CIE;

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
```



papergrid

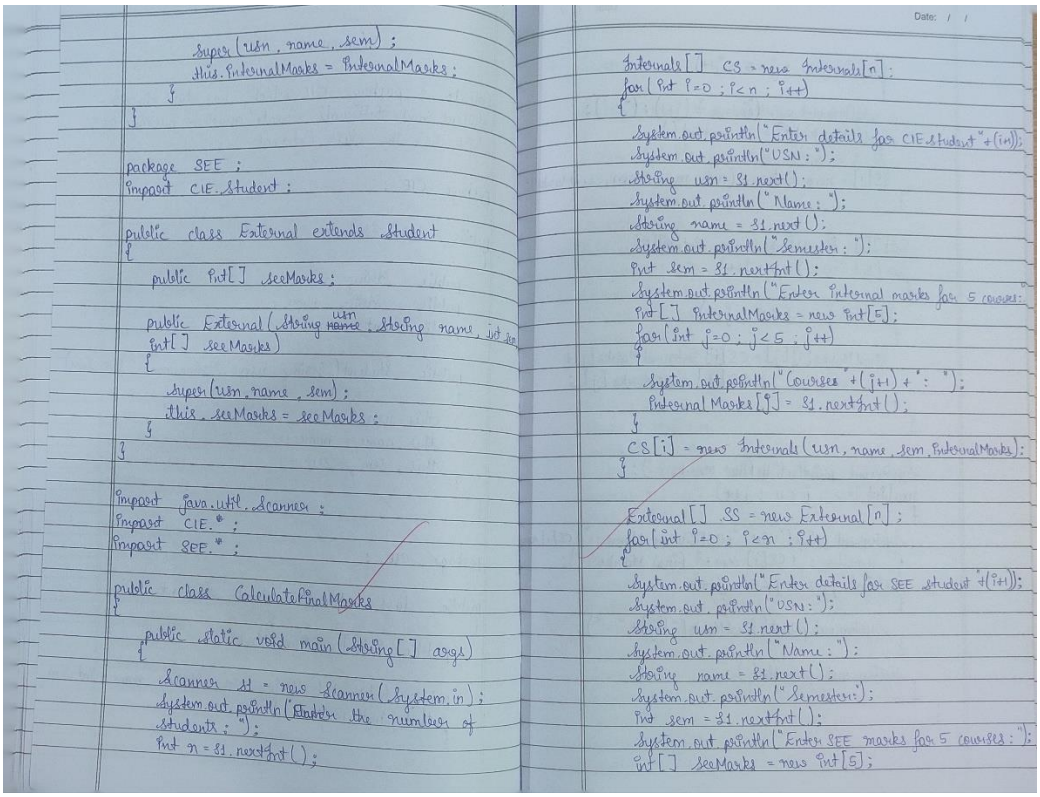
Date: / /

22.1.24

LABORATORY - 6

Create a package CIE which has two classes - Student and Internals. Create another package SEE which has the class External

```
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 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 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 si = new Scanner ( System . in ) ;
        System . out . println ( "Enter the number of
        students : ") ;
        int n = si . nextInt ( ) ;
```

Date: / /

```
Internals[ ] CS = new Internals[n] ;
for ( int i=0 ; i<n ; i++)
{
    System . out . println ( "Enter details for CIE student " + (i+1)) ;
    System . out . println ( "USN : ") ;
    String usn = si . next ( ) ;
    System . out . println ( "Name : ") ;
    String name = si . next ( ) ;
    System . out . println ( "Semester : ") ;
    int sem = si . nextInt ( ) ;
    System . out . println ( "Enter Internal marks for 5 courses: ") ;
    int[ ] InternalMarks = new int[5] ;
    for ( int j=0 ; j<5 ; j++)
    {
        System . out . println ( "Courses " + (j+1) + " : ") ;
        InternalMarks[j] = si . 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 . println ( "USN : ") ;
    String usn = si . next ( ) ;
    System . out . println ( "Name : ") ;
    String name = si . next ( ) ;
    System . out . println ( "Semester : ") ;
    int sem = si . nextInt ( ) ;
    System . out . println ( "Enter SEE marks for 5 courses : ") ;
    int[ ] seeMarks = new int[5] ;
```

```
for(int j=0 ; j<5 ; j++)
{
  System.out.println("Course "+(j+1)+(": ");
  seeMarks[j] = sc.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("\n Final Marks : ");
for(int i=0 ; j<n ; j++)
{
  System.out.print("USN : "+CS[i].usn+", Name : "+CS[i].n
  +" Semester : "+CS[i].sem+", Final Marks : ");
  for(int j=0 ; j<5 ; j++)
  {
    System.out.print(finalMarks[i][j]+" ");
  }
  System.out.println();
}
}
```

OUTPUT :

```
Enter the number of students :
2
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 : 46
Course 4 : 49
Course 5 : 50
Enter details for CIE student 2
USN : 2
Name : Krishna
Semester : 3
Enter Internal marks for 5 courses :
Course 1 : 47
Course 2 : 47
Course 3 : 50
Course 4 : 48
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 : 50
Course 4 : 50
Course 5 : 49
Enter details for SEE student 2
USN : 2
Name : Krishna
Semester : 3
Enter SEE marks for 5 courses :
Course 1 : 47
Course 2 : 49
Course 3 : 49
Course 4 : 50
Course 5 : 48

Final Marks :

USN : 1 , Name : Ram , Semester : 3 , Final Marks :
95 97 96 99 99
USN : 2 , Name : Krishna , Semester : 3 , Final Marks :
96 96 99 96 97
```

LABORATORY PROGRAM - 7

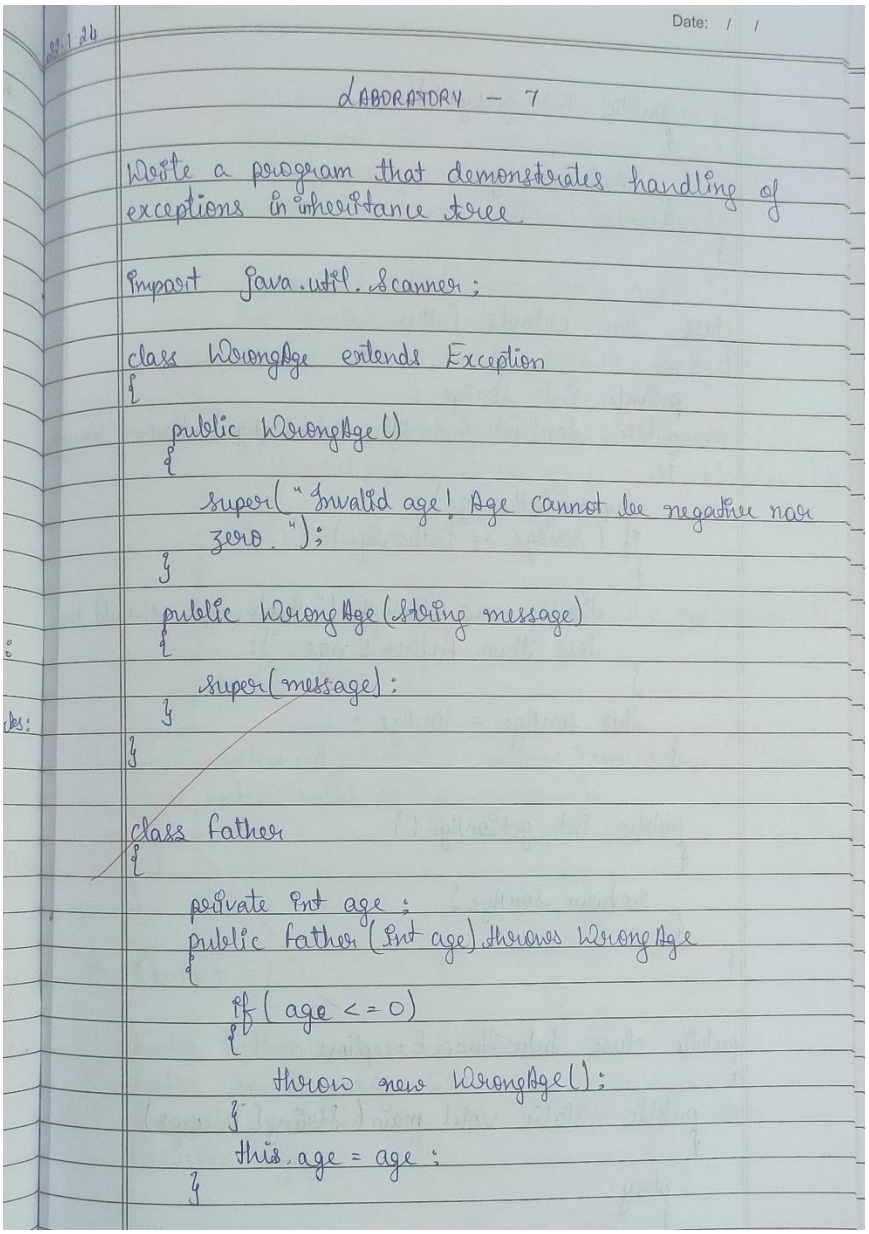
Write a program that demonstrates handling of exceptions in inheritance tree. Create a base class called “Father” and derived class called “Son” which extends the base class. In Father class, implement a constructor which takes the age and throws the exception WrongAge( ) when the input age<0. In Son 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

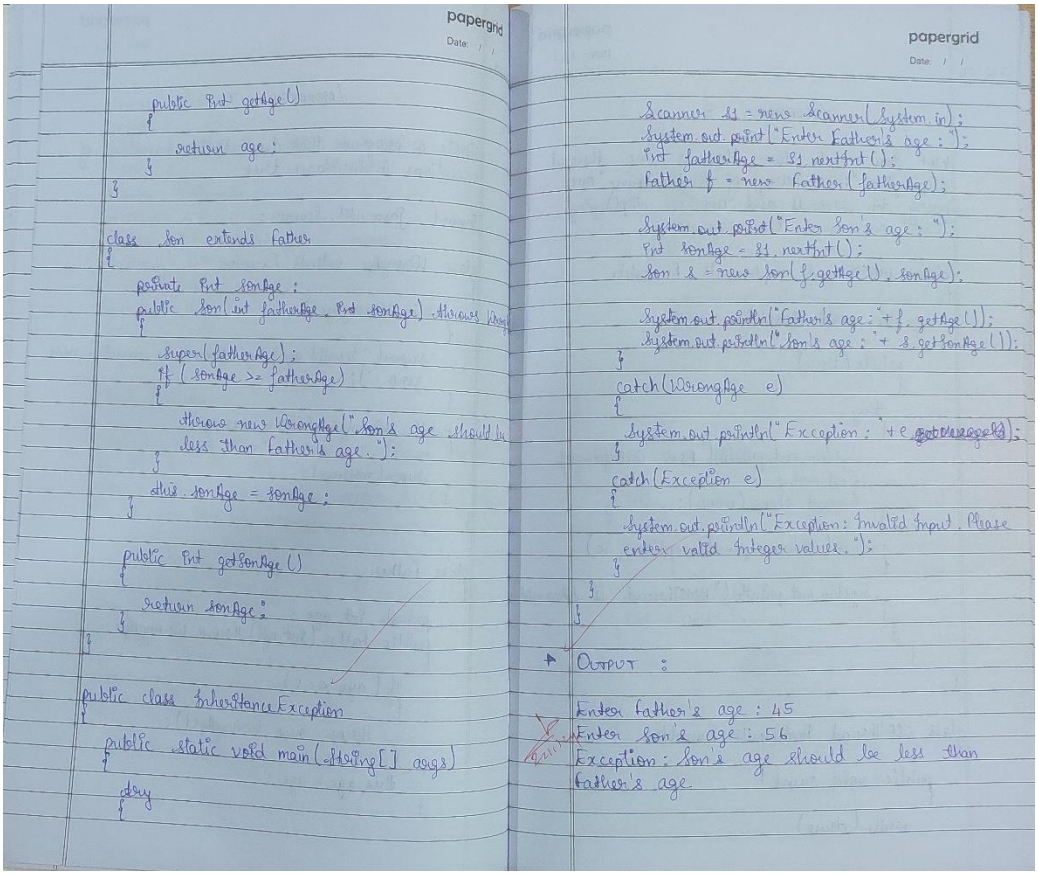
```
D:\NotePad++\Java>javac InheritanceException.java

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
```



LABORATORY - 7

Write a program that demonstrates handling of exceptions in inheritance tree.

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

            System.out.print("Enter Son's age : ");
            int sonAge = ss.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.getMessage());
        }
        catch(Exception e)
        {
            System.out.println("Exception: Invalid Input. Please
            enter valid integer values.");
        }
    }
}
```

OUTPUT :

```
Enter father's age : 45
Enter Son's age : 56
Exception : Son's age should be less than
father's age.
```

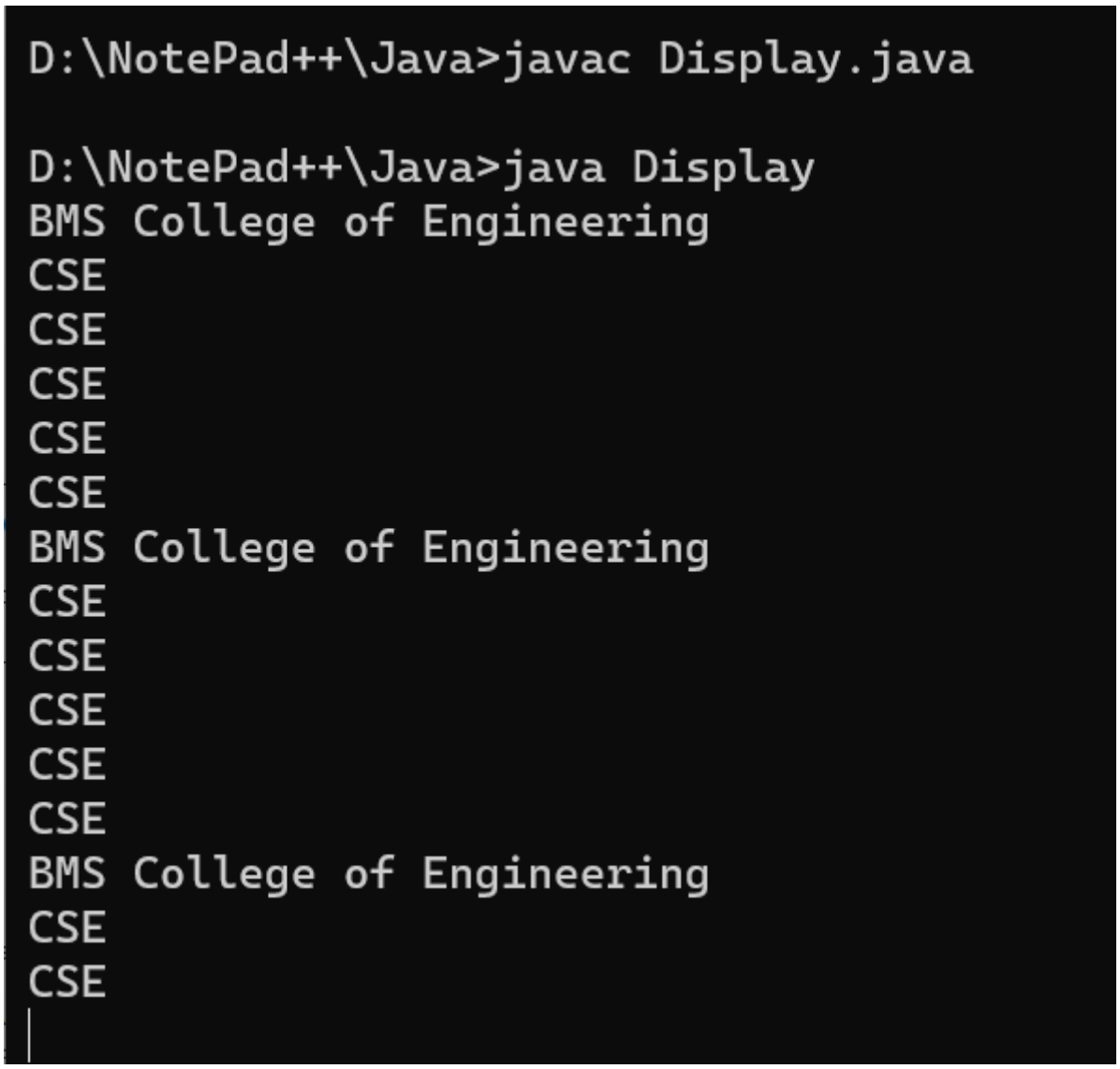
LABORATORY PROGRAM - 8

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



5.2.24

LABORATORY - 8

WAP that create 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 :

```
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
```

5/2/24

LABORATORY PROGRAM - 9

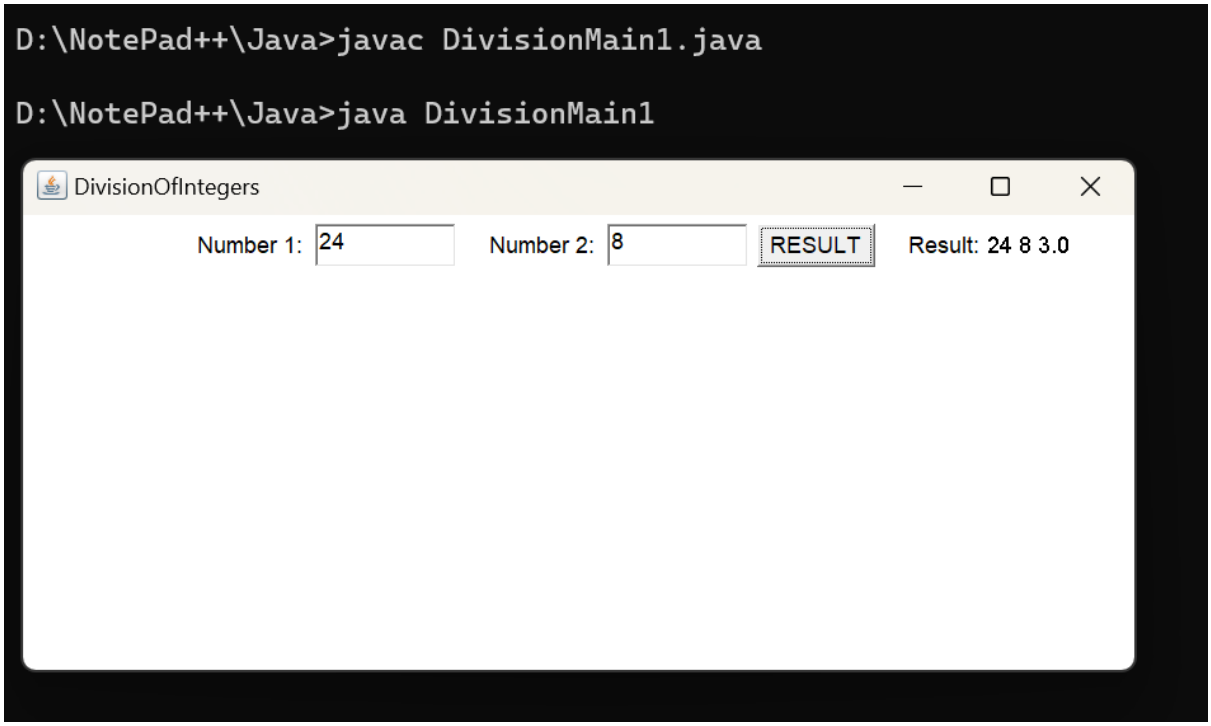
Write a program that creates a user interface to perform integer divisions. The user enters two numbers in the text fields, Num1 and Num2. The division of Num1 and Num2 is displayed in the Result field when the Divide button is clicked. If Num1 or Num2 were not an integer, the program would throw a NumberFormatException.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



19.2.24

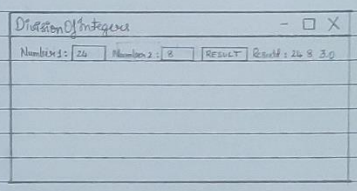
papergrid
Date: / /

LABORATORY - 9

WAP that creates a user interface to perform integer divisions.

```
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());
                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("Division of Integers");
        dm.setVisible(true);
    }
}
```

OUTPUT :



13.2.24

LABORATORY - 10

REPORT

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

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

iv) DivisionMain : It 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.

papergrid
Date: / /

v) DivisionMaint : It another frame-based Java app that performs division operations similar to DivisionMain. However, it handles exceptions in a different order compared to DivisionMain.

vi) TextFieldDemo : Demonstrates the usage of the text fields in Java AWT. It provides a simple GUI interface where users can input their name and password. Upon pressing Enter in either text field, the program updates the window to display the entered name and password.

vii) ButtonDlgTD : It a frame-based Java app that demonstrates event handling and dialog creation. It consists of 3 buttons labeled "Yes", "No", and third "Undecided". Clicking on any button opens a dialog window displaying the button label.

In conclusion, the 6 mentioned Java programs exemplify fundamental GUI application development and event handling techniques.

papergrid
Date: / /

REPORT ON MOUSEEVENTSDEMO PROGRAM

This showcases the implementation of mouse event handling in Java. It provides a simple graphical user interface where users can interact with the mouse and the program responds to various mouse events.

In conclusion, this program, through its intuitive interface and dynamic feedback, users can understand and interact with different mouse actions.