

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



LAB REPORT
on
Object Oriented Java Programming
(23CS3PCOOJ)

Submitted by

K.Keerthi Reddy(**1BM23CS137**)

in partial fulfillment for the award of the degree of
BACHELOR OF ENGINEERING
in
COMPUTER SCIENCE AND ENGINEERING



B.M.S. COLLEGE OF ENGINEERING
(Autonomous Institution under VTU)
BENGALURU-560019

Sep-2024 to Jan-2025

**B.M.S. College of Engineering,
Bull Temple Road, Bangalore 560019
(Affiliated To Visvesvaraya Technological University, Belgaum)
Department of Computer Science and Engineering**



CERTIFICATE

This is to certify that the Lab work entitled “Object Oriented Java Programming (23CS3PCOOJ)” carried out by **K.Keerthi Reddy(1BM23CS137)**, who is bonafide student of **B.M.S. College of Engineering**. It is in partial fulfillment for the award of **Bachelor of Engineering in Computer Science and Engineering** of the Visvesvaraya Technological University, Belgaum. The Lab report has been approved as it satisfies the academic requirements in respect of an Object Oriented Java Programming (23CS3PCOOJ) work prescribed for the said degree.

Lab faculty Incharge Name Assistant Professor Department of CSE, BMSCE	Dr. Jyothi S Nayak Professor & HOD Department of CSE, BMSCE
--	---

Index

Sl. No.	Date	Experiment Title	Page No.
1	30/9/24	Quadratic equations	1
2	7/10/24	Calculation of SGPA	3
3	14/10/24	Bookdetails using <code>toString()</code>	5
4	21/10/24	Shape and areas (Abstract classes)	6
5	11/10/24	Bank class using Inheritance	8
6	28/11/24	Marks card of student using package	13
7	28/11/24	Father and son's age using exception handling	16
8	28/11/24	Display college name & department using threads	18
9	28/11/24	Creation of divider add	22
10	28/11/24	A) Demonstration of interprocess communication B) Demonstration of deadlock	19 21

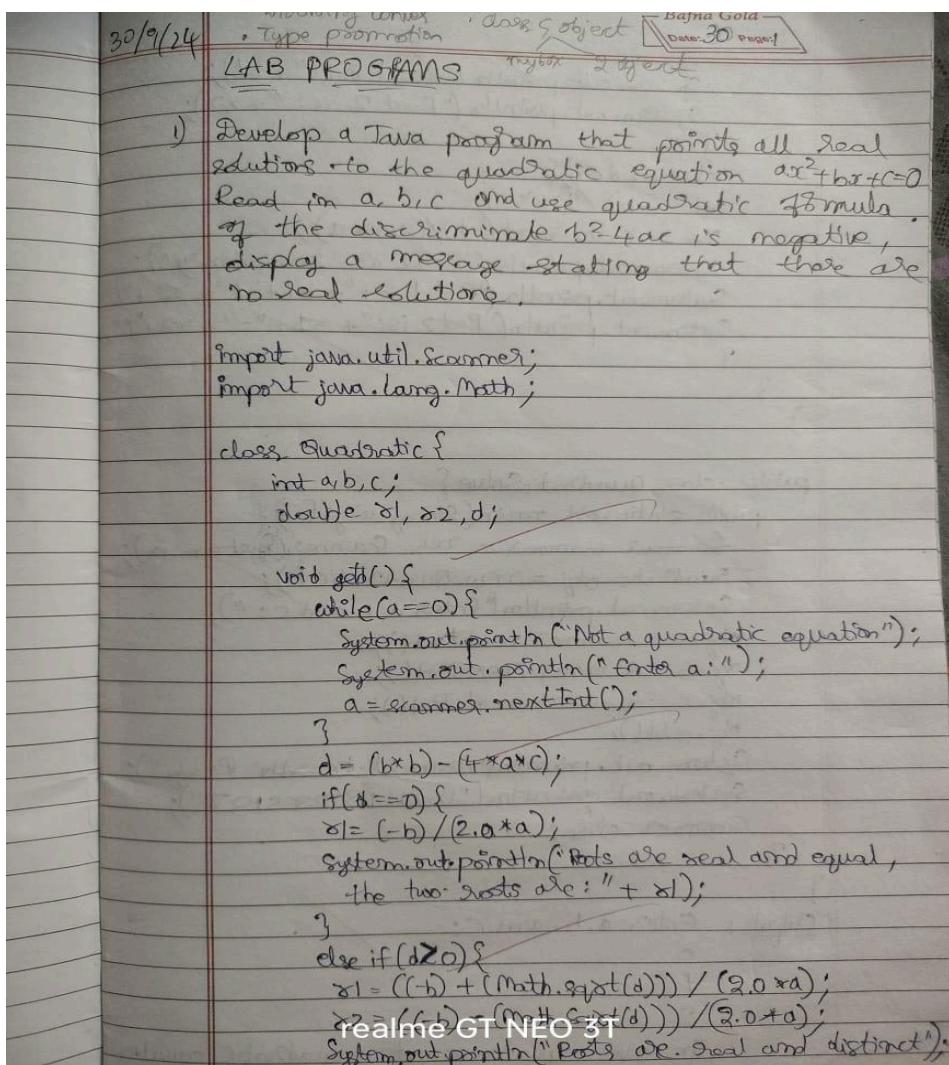
Github Link:

<https://github.com/keerthireddy7272/Java-Lab-programs>

Program 1

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

Algorithm:



```

System.out.println("Root 1 is: " + r1);
System.out.println("Root 2 is: " + r2);
}
else if(d < 0){
    r1 = (-b) / (2.0 * a);
    r2 = Math.sqrt(-d) / (2.0 * a);
    System.out.println("Roots are imaginary");
    System.out.println("Root 1 is: " + r1 + " + " + r2 + "i");
    System.out.println("Root 2 is: " + r1 + " - " + r2 + "i");
}
}
}

```

```

public class QuadraticSolve {
    public static void main(String args[]) {
        Scanner scanner = new Scanner(System.in);
        Quadratic obj = new Quadratic();
        System.out.println("Enter a, b and c: ");
        obj.a = scanner.nextInt();
        obj.b = scanner.nextInt();
        obj.c = scanner.nextInt();
        obj.get();
        System.out.println("Name: K. Keerthi Reddy");
        System.out.println("USN: 1BMRSCS137");
        scanner.close();
    }
}

```

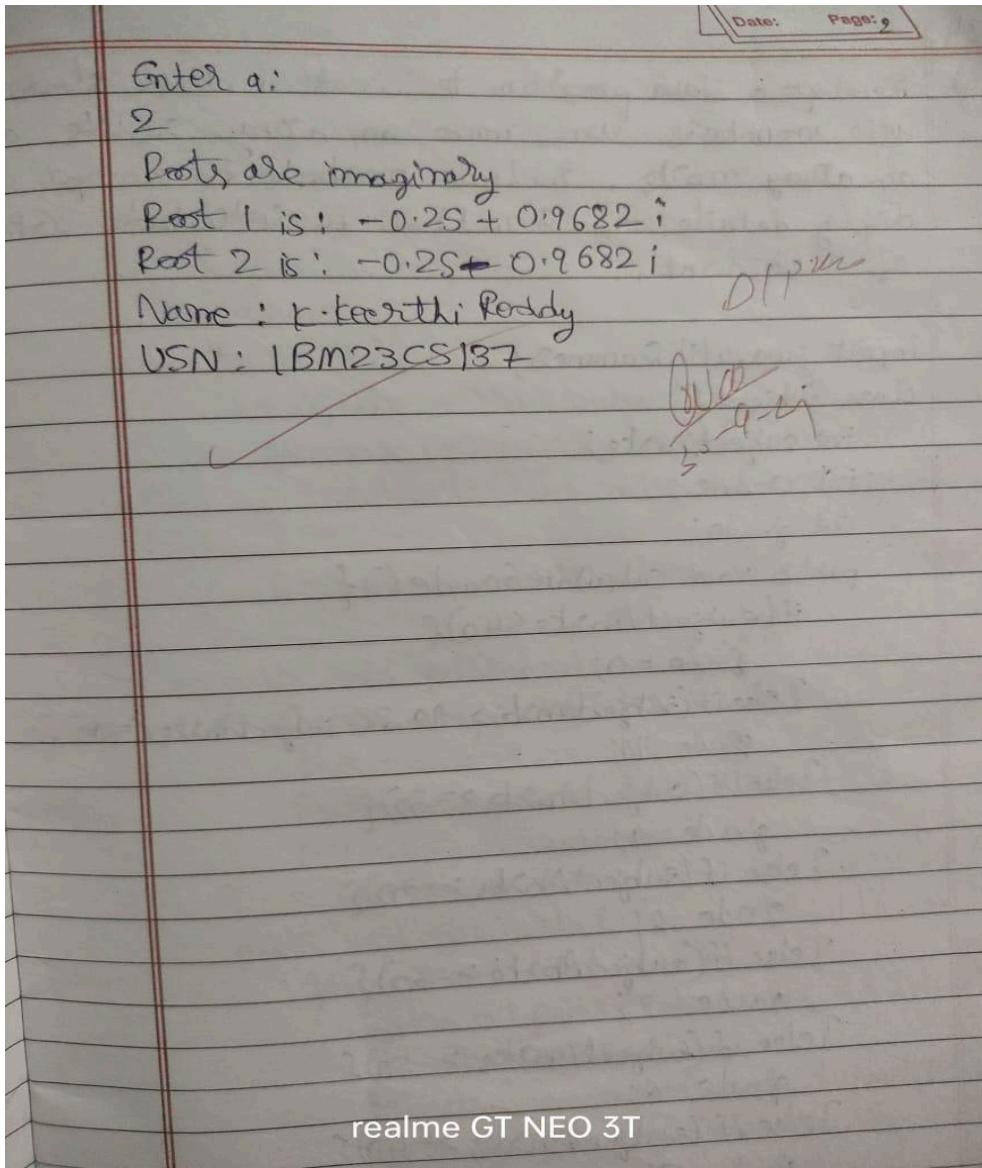
Output: Enter a, b, and C:

0

1

2

The given equation is not a quadratic equation
realme GT NEO 3T



Code:

```
# Java-Lab-programs
import java.util.Scanner;
import java.lang.Math;

class Quadratic {
    int a, b, c;
    double r1, r2, d;

    void getd() {
        while (a == 0) {
            System.out.println("The given equation is not a quadratic equation");
            System.out.println("Enter a:");
            Scanner scanner = new Scanner(System.in);
            a = scanner.nextInt();
```

```

}
d = (b * b) - (4 * a * c);

if (d == 0) {
r1 = (-b) / (2.0 * a);
System.out.println("Roots are real and equal, the two roots are: " + r1);
}

else if (d > 0) {
r1 = ((-b) + (Math.sqrt(d))) / (2.0 * a);
r2 = ((-b) - (Math.sqrt(d))) / (2.0 * a);
System.out.println("Roots are real and distinct");
System.out.println("Root 1 is: " + r1);
System.out.println("Root 2 is: " + r2);
}
else if(d<0){
r1 = (-b) / (2.0 * a);
r2 = Math.sqrt(-d) / (2.0 * a);
System.out.println("Roots are imaginary");
System.out.println("Root 1 is: " + r1 + " + " + r2 + "i");
System.out.println("Root 2 is: " + r1 + " - " + r2 + "i");
}
}

public class QuadraticSolve {
public static void main(String args[]) {
Scanner scanner = new Scanner(System.in);
Quadratic obj = new Quadratic();
System.out.println("Enter a, b, and c: ");
obj.a = scanner.nextInt();
obj.b = scanner.nextInt();
obj.c = scanner.nextInt();
obj.getd();
System.out.println("Name: K. Keerthi Reddy");
System.out.println("USN: 1BM23CS137");
scanner.close();
}
}
a = scanner.nextInt();
}
d = (b * b) - (4 * a * c);

if (d == 0) {
r1 = (-b) / (2.0 * a);
System.out.println("Roots are real and equal, the two roots are: " + r1);
}

```

```

else if(d > 0) {
    r1 = ((-b) + (Math.sqrt(d))) / (2.0 * a);
    r2 = ((-b) - (Math.sqrt(d))) / (2.0 * a);
    System.out.println("Roots are real and distinct");
    System.out.println("Root 1 is: " + r1);
    System.out.println("Root 2 is: " + r2);
}
else if(d<0){
    r1 = (-b) / (2.0 * a);
    r2 = Math.sqrt(-d) / (2.0 * a);
    System.out.println("Roots are imaginary");
    System.out.println("Root 1 is: " + r1 + " + " + r2 + "i");
    System.out.println("Root 2 is: " + r1 + " - " + r2 + "i");
}
}
}

public class QuadraticSolve {
public static void main(String args[]) {
    Scanner scanner = new Scanner(System.in);
    Quadratic obj = new Quadratic();
    System.out.println("Enter a, b, and c: ");
    obj.a = scanner.nextInt();
    obj.b = scanner.nextInt();
    obj.c = scanner.nextInt();
    obj.getd();
    System.out.println("Name: K. Keerthi Reddy");
    System.out.println("USN: 1BM23CS137");
    scanner.close();
}
}

```

Output:

```

import java.util.Scanner;
import java.lang.Math;

class Quadratic {
int a, b, c;
double r1, r2, d;

void getd() {
while (a == 0) {
System.out.println("The given equation is not a quadratic equation");
System.out.println("Enter a:");
a = scanner.nextInt();
}
d = (b * b) - (4 * a * c);

if (d == 0) {
r1 = (-b) / (2.0 * a);
System.out.println("Roots are real and equal, the two roots are: " + r1);
}

else if (d > 0) {
r1 = ((-b) + (Math.sqrt(d))) / (2.0 * a);
r2 = ((-b) - (Math.sqrt(d))) / (2.0 * a);
System.out.println("Roots are real and distinct");
System.out.println("Root 1 is: " + r1);
System.out.println("Root 2 is: " + r2);
}
else if (d < 0) {
r1 = (-b) / (2.0 * a);
r2 = Math.sqrt(-d) / (2.0 * a);
System.out.println("Roots are imaginary");
System.out.println("Root 1 is: " + r1 + " + " + r2 + "i");
System.out.println("Root 2 is: " + r1 + " - " + r2 + "i");
}
}
}

public class QuadraticSolve {
public static void main(String args[]) {
Scanner scanner = new Scanner(System.in);
Quadratic obj = new Quadratic();
System.out.println("Enter a, b, and c: ");
obj.a = scanner.nextInt();
obj.b = scanner.nextInt();
obj.c = scanner.nextInt();
obj.getd();
System.out.println("Name: K. Keerthi Reddy");
System.out.println("USN: 1BM23CS137");
scanner.close();
}
}

```

```

Root 2 is: -1.0
Name: K. Keerthi Reddy
USN: 1BM23CS137

D:\1BM23CS137>java QuadraticSolve
Enter a, b, and c:
1
2
8
Roots are imaginary
Root 1 is: -1.0 + 2.6457513110645907i
Root 2 is: -1.0 - 2.6457513110645907i
Name: K. Keerthi Reddy
USN: 1BM23CS137

D:\1BM23CS137>java QuadraticSolve
Enter a, b, and c:
0
1
2
The given equation is not a quadratic equation
Enter a:
2
Roots are imaginary
Root 1 is: -0.25 + 0.9682458365518543i
Root 2 is: -0.25 - 0.9682458365518543i
Name: K. Keerthi Reddy
USN: 1BM23CS137

D:\1BM23CS137>

```

Program 2

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

Algorithm:

7/10/24 *array of objects*

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

realme GT NEO 3T

Bajna Gola
Date: Page: 3

```

class Student {
    String name;
    String uen;
    double CGPA;
    Subject[] subjects;
    Scanner s;
    Student() {
        subjects = new Subject[8];
        for (int i=0; i<8; i++) {
            subjects[i] = new Subject();
        }
        s = new Scanner(System.in);
    }
    public void getStudentDetails() {
        System.out.print("Enter student name: ");
        this.name = s.nextLine();
        System.out.print("Enter student uen: ");
        this.uen = s.nextLine();
    }
    public void getMarks() {
        for (int i=0; i<8; i++) {
            System.out.print("Enter marks for subject " +
                (i+1) + ": ");
            subjects[i].subjectMarks = s.nextInt();
            System.out.print("Enter credits for subject " +
                (i+1) + ": ");
            subjects[i].credits = s.nextInt();
            subjects[i].calculateGrade();
        }
        System.out.println();
    }
}
realme GT NEO 3T

```

```

public void computeSGPA() {
    double totalPoints = 0;
    int totalCredits = 0;
    for (int i=0; i<8; i++) {
        totalPoints += subjects[i].grade *
            subjects[i].credits;
        totalCredits += subjects[i].credits;
    }
    SGPA = (totalCredits == 0) ? 0 : totalPoints / totalCredits;
}

```

```

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

```

```

public class Student2 {
    public static void main (String [] args) {
        Student s1 = new Student1();
        s1.getStudentDetails();
        s1.getMarks();
        s1.computeSGPA();
        s1.displayResults();
    }
}

```

realme GT NEO 3T

Buf Date Gold Page: 4

Output:

Enter student name : Keerthi Reddy
 Enter student USN : 1BM23CS137
 Enter marks for subject 1 : 90
 Enter credits for Subject 1 : 4
 Enter marks for subject 2 : 90
 Enter credits for subject 2 : 4
 Enter marks for subject 3 : 85
 Enter credits for subject 3 : 3
 Enter marks for subject 4 : 82
 Enter credits for subject 4 : 3
 Enter marks for subject 5 : 87
 Enter credits for subject 5 : 3
 Enter marks for subject 6 : 95
 Enter credits for subject 6 : 1
 Enter marks for subject 7 : 72
 Enter credits for subject 7 : 1
 Enter marks for subject 8 : 91
 Enter credits for subject 8 : 1
 Student Name: Keerthi Reddy
 USN: 1BM23CS137
 SGPA: 9.45

realme GT NEO 3T

Code:

```
import java.util.Scanner;
```

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

    public void calculateGrade(){
        if(subjectMarks < 40) {
            grade = 0;
        } else if (subjectMarks >= 90 && subjectMarks <= 100) {
            grade = 10;
        } else if (subjectMarks >= 80) {
```

```

        grade = 9;
    } else if (subjectMarks >= 70){
        grade = 8;
    } else if (subjectMarks >= 60){
        grade = 7;
    } else if (subjectMarks >= 50){
        grade = 6;
    } else if (subjectMarks >= 40){
        grade = 5;
    } else{
        grade = 0;
    }
}

class Student1 {
    String name;
    String usn;
    double SGPA;
    Subject[] subjects;
    Scanner s;

    Student1() {
        subjects = new Subject[8];
        for (int i = 0; i < 8; i++) {
            subjects[i] = new Subject();
        }
        s = new Scanner(System.in);
    }

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

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

```

```
public void computeSGPA() {  
    double totalPoints = 0;  
    int totalCredits = 0;  
    for (int i = 0; i < 8; i++) {  
        totalPoints += subjects[i].grade * subjects[i].credits;  
        totalCredits += subjects[i].credits;  
    }  
    SGPA = (totalCredits == 0) ? 0 : totalPoints / totalCredits;  
}  
public void displayResults() {  
    System.out.println("Student Name: " + this.name);  
    System.out.println("USN: " + this.usn);  
    System.out.printf("SGPA: %.2f%n", SGPA);  
}  
}  
public class Student2 {  
    public static void main(String[] args) {  
        Student1 s1 = new Student1();  
        s1.getStudentDetails();  
        s1.getMarks();  
        s1.computeSGPA();  
        s1.displayResults();  
    }  
}
```

Output:

```
import java.util.Scanner;

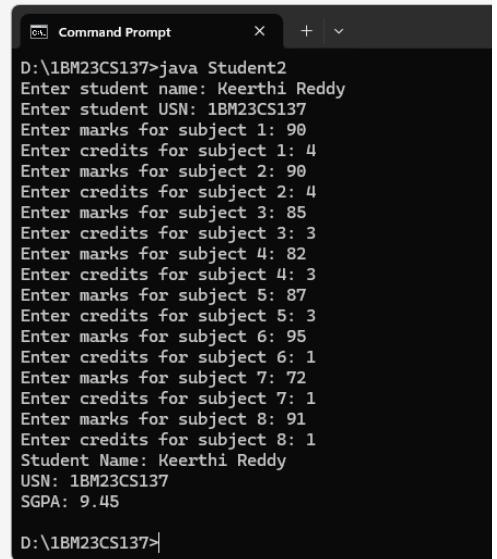
class Subject{
    int subjectMarks;
    int credits;
    int grade;

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

class Student1 {
    String name;
    String usn;
    double SGPA;
    Subject[] subjects;
    Scanner s;

    Student1() {
        subjects = new Subject[8];
        for (int i = 0; i < 8; i++) {
            subjects[i] = new Subject();
        }
        s = new Scanner(System.in);
    }

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



D:\1BM23CS137>java Student2
Enter student name: Keerthi Reddy
Enter student USN: 1BM23CS137
Enter marks for subject 1: 90
Enter credits for subject 1: 4
Enter marks for subject 2: 90
Enter credits for subject 2: 4
Enter marks for subject 3: 85
Enter credits for subject 3: 3
Enter marks for subject 4: 82
Enter credits for subject 4: 3
Enter marks for subject 5: 87
Enter credits for subject 5: 3
Enter marks for subject 6: 95
Enter credits for subject 6: 1
Enter marks for subject 7: 72
Enter credits for subject 7: 1
Enter marks for subject 8: 91
Enter credits for subject 8: 1
Student Name: Keerthi Reddy
USN: 1BM23CS137
SGPA: 9.45
D:\1BM23CS137>

Program 3

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

Algorithm:

14/10/24

Create a class Book which contains 4 members : name, author, num_pages. Include a constructor to set the values for members. Include methods to set and get the details of the objects. Include a toString() method that could display complete details of the book. Develop a java program to create n book objects .

```
import java.util.Scanner;
class Book {
    String name;
    String author;
    int price;
    int numPages;

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

    public String toString() {
        return "The name of book is: " + this.name + "\n" +
               "Author name: " + this.author + "\n" +
               "Price of book is: " + this.price + "\n" +
               "Number of pages: " + this.numPages + "\n";
    }

    public class Bookdetails {
        public static void main (String [] args) {
            Scanner sc = new Scanner (System.in);
            realme GT NEO 3T
        }
    }
}
```

```
System.out.print("Enter number of books: ");  
int n = s.nextInt();  
Books[] b = new Books[n];
```

```
for (int i=0; i<n; i++) {  
    System.out.print("Enter name of book: ");  
    s.nextLine();  
    String name = s.nextLine();  
    System.out.print("Enter author of book: ");  
    s.nextLine();  
    String author = s.nextLine();  
    System.out.print("Enter price of book: ");  
    int price = s.nextInt();  
    System.out.print("Enter no. of pages of book: ");  
    int numPages = s.nextInt();  
    b[i] = new Books(name, author, price, numPages);
```

```
}  
System.out.println("\nBook Details: ");  
for (int i=0; i<n; i++) {  
    System.out.println(b[i].toString());
```

```
}  
System.out.println("Name : K. Keerthi Reddy");  
System.out.println("USN: KBM23CS137");  
s.close();
```

Output :

Enter number of books : 2

Enter name of the book : harry potter

Enter auths of book : J.K. Rowling

Enter price of book : 5690

Enter pages of the book : 500

Enter name of the book : cursed child

Enter author of book : J.K. Rowling

Enter price of book : 7095

Enter pages of book : 600

Book Details:

The name of book is : harry potter

Auths name : J.K. Rowling

The price of book is : 5690

Number of pages : 500

The name of book is : Cursed child

Auths name : J.K. Rowling

The price of book is : 7095

Number of pages : 600

Name of student : R. Keerthi Reddy

USN of the student : 1BM23CS137

R
11.10

realme GT NEO 3T

Code:

```
import java.util.Scanner;

class Books {
    String name;
    String author;
    int price;
    int numPages;

    Books(String name, String author, int price, int numPages) {
        this.name = name;
        this.author = author;
        this.price = price;
        this.numPages = numPages;
    }
    public String toString() {
        return "The name of the book is : " + this.name + "\n" +
               "Author name: " + this.author + "\n" +
               "The price of the book is: " + this.price + "\n" +
               "Number of pages : " + this.numPages + "\n";
    }
}

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

        System.out.print("Enter the number of books: ");
        int n = s.nextInt();
        Books[] b = new Books[n];

        for (int i = 0; i < n; i++) {
            System.out.print("Enter name of the book: ");
            s.nextLine();
            String name = s.nextLine();
            System.out.print("Enter author of the book: ");
            String author = s.nextLine();
            System.out.print("Enter the price of the book: ");
            int price = s.nextInt();
            System.out.print("Enter the number of pages of the book: ");
            int numPages = s.nextInt();

            b[i] = new Books(name, author, price, numPages);
        }

        System.out.println("\nBook Details:");
        for (int i = 0; i < n; i++) {
```

```

        System.out.println(b[i].toString());
    }
    System.out.println("Name of the student: K. Keerthi Reddy");
    System.out.println("USN of the student: 1BM23CS137");
    s.close();
}
}

```

Output:

```

import java.util.Scanner;
class Books {
    String name;
    String author;
    int price;
    int numPages;

    Books(String name, String author, int price, int numPages) {
        this.name = name;
        this.author = author;
        this.price = price;
        this.numPages = numPages;
    }

    public String toString() {
        return "The name of the book is : " + this.name + "\n" +
            "Author name: " + this.author + "\n" +
            "The price of the book is: " + this.price + "\n" +
            "Number of pages : " + this.numPages + "\n";
    }
}

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

        System.out.print("Enter the number of books: ");
        int n = s.nextInt();
        Books[] b = new Books[n];

        for (int i = 0; i < n; i++) {
            System.out.print("Enter name of the book: ");
            s.nextLine();
            String name = s.nextLine();
            System.out.print("Enter author of the book: ");
            String author = s.nextLine();
            System.out.print("Enter the price of the book: ");
            int price = s.nextInt();
            System.out.print("Enter the number of pages of the book: ");
            int numPages = s.nextInt();

            b[i] = new Books(name, author, price, numPages);
        }

        System.out.println("\nBook Details:");
        for (int i = 0; i < n; i++) {
            System.out.println(b[i].toString());
        }
        System.out.println("Name of the student: K. Keerthi Reddy");
        System.out.println("USN of the student: 1BM23CS137");
        s.close();
    }
}

```

```

D:\1BM23CS137>java Bookdetails
Enter the number of books: 2
Enter name of the book: harry potter
Enter author of the book: J.K.Rowling
Enter the price of the book: 5690
Enter the number of pages of the book: 500
Enter name of the book: cursed child
Enter author of the book: J.K.Rowling
Enter the price of the book: 7095
Enter the number of pages of the book: 600

Book Details:
The name of the book is : harry potter
Author name: J.K.Rowling
The price of the book is: 5690
Number of pages : 500

The name of the book is : cursed child
Author name: J.K.Rowling
The price of the book is: 7095
Number of pages : 600

Name of the student: K. Keerthi Reddy
USN of the student: 1BM23CS137

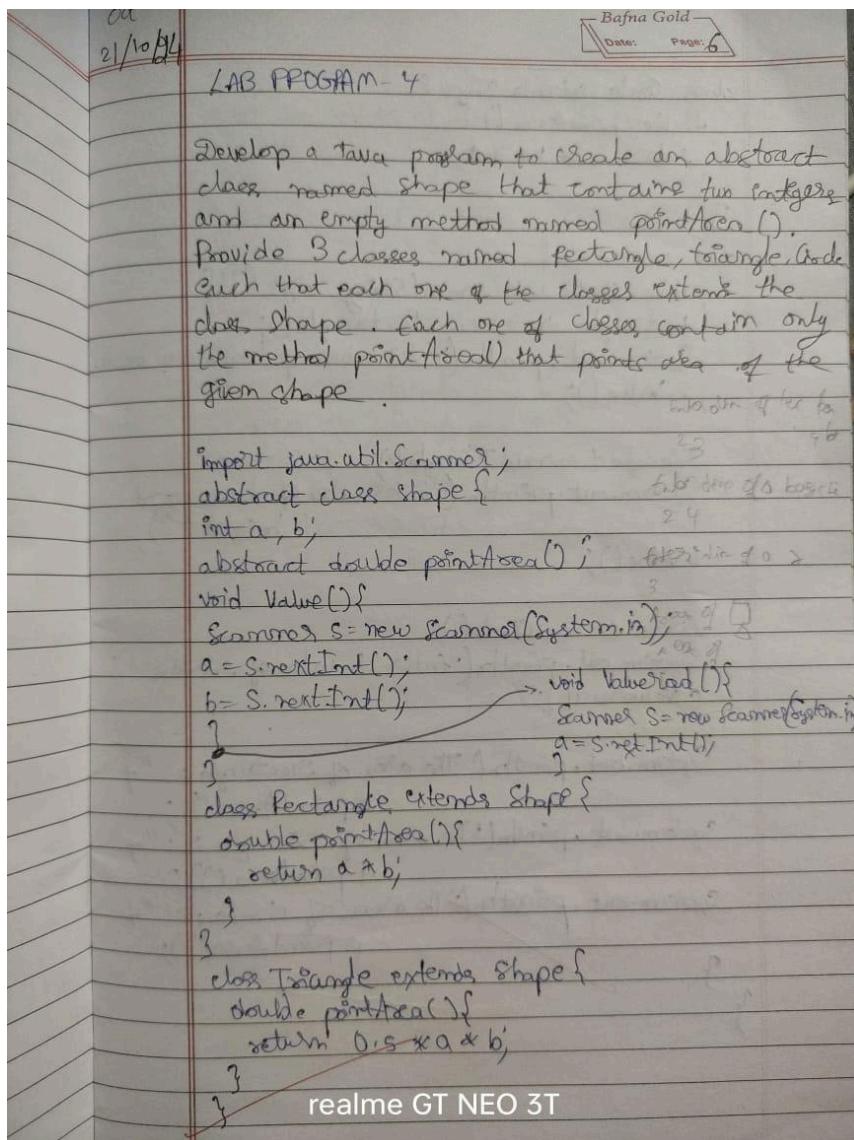
```

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.

Algorithm:



```

class Circle extends Shape {
    double printArea () {
        return 3.14 * a * a;
    }
}

class ShapeArea {
    public static void main (String args []) {
        Rectangle r = new Rectangle ();
        System.out.println ("Enter values of length and breadth:");
        r.Value ();
    }

    Triangle t = new Triangle ();
    System.out.println ("Enter values of base and height:");
    t.Value ();

    Circle c = new Circle ();
    System.out.println ("Enter the value of radius:");
    c.ValueRadius ();
}

System.out.println ("The area of rectangle is: " + r.printArea ());
System.out.println ("The area of triangle is: " + t.printArea ());
System.out.println ("The area of circle is: " + c.printArea ());

}
}

```

realme GT NEO 3T

Output

Enter the values of length and breadth :

2

3

Enter the values of base and height :

5

2

Enter the value of radius :

2

The area of rectangle is : 6.0

The area of Triangle is : 5.0

The area of Circle is : 12.56

Name : K. Keerthi Reddy

USN : 1BM23CS137

21.10

Code:

```
import java.util.Scanner;

abstract class Shape{
int a,b;
abstract double printArea();
void Value(){
Scanner s = new Scanner(System.in);
a=s.nextInt();
b=s.nextInt();
}
void Valuerad(){
Scanner s = new Scanner(System.in);
a=s.nextInt();
}
}

class Rectangle extends Shape{
double printArea(){
return a*b;
}
}

class Triangle extends Shape{
double printArea(){
return 0.5 * a * b;
}
}

class Circle extends Shape{
double printArea(){
return 3.14 * a * a;
}
}

class shapeArea{
public static void main(String args[]){
Rectangle r = new Rectangle();
System.out.println("Enter the values of length and breadth: ");
r.Value();

Triangle t = new Triangle();
System.out.println("Enter the values of base and height: ");
t.Value();

Circle c = new Circle();
System.out.println("Enter the value of radius: ");
c.Valuerad();
}
```

```

System.out.println("The area of rectangle is: " + r.printArea());
System.out.println("The area of Triangle is: " + t.printArea());
System.out.println("The area of Circle is: " + c.printArea());
System.out.println("Name: K.Keerthi Reddy");
System.out.println("Usn : 1BM23CS137");
}
}
}

```

Output:

```

import java.util.Scanner;
abstract class Shape{
int a,b;
abstract double printArea();
void Value(){
Scanner s = new Scanner(System.in);
a=s.nextInt();
b=s.nextInt();
}
void Valuerad(){
Scanner s = new Scanner(System.in);
a=s.nextInt();
}
}

class Rectangle extends Shape{
double printArea(){
return a*b;
}
}
class Triangle extends Shape{
double printArea(){
return 0.5 * a *b;
}
}
class Circle extends Shape{
double printArea(){
return 3.14 * a * a;
}
}

class shapeArea{
public static void main(String args[]){
Rectangle r = new Rectangle();
System.out.println("Enter the values of length and breadth: ");
r.Value();

Triangle t = new Triangle();
System.out.println("Enter the values of base and height: ");
t.Value();
Circle c = new Circle();
System.out.println("Enter the value of radius: ");
c.Valuerad();

System.out.println("The area of rectangle is: " + r.printArea());
System.out.println("The area of Triangle is: " + t.printArea());
System.out.println("The area of Circle is: " + c.printArea());
System.out.println("Name: K.Keerthi Reddy");
System.out.println("Usn : 1BM23CS137");
}
}

```

```

D:\1BM23CS137>javac shapeArea.java
D:\1BM23CS137>java shapeArea
Enter the values of length and breadth:
2
3
Enter the values of base and height:
5
2
Enter the value of radius same number twice:
2
The area of rectangle is: 6.0
The area of Triangle is: 5.0
The area of Circle is: 12.56
D:\1BM23CS137>

D:\1BM23CS137>javac shapeArea
D:\1BM23CS137>java shapeArea
Enter the values of length and breadth:
2
3
Enter the values of base and height:
5
2
Enter the value of radius:
2
The area of rectangle is: 6.0
The area of Triangle is: 5.0
The area of Circle is: 12.56
Name: K.Keerthi Reddy
Usn : 1BM23CS137
D:\1BM23CS137>

```

Program 5

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.

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
- e) Check for the minimum balance, impose penalty if necessary and update the balance.

Algorithm:

28/10/24
LAB PROGRAM - 5

Bank	
Savings Acc	Current Acc
1) Cont	1) No. of days
2) Withdrawal	2) Checkbook

class Bank {
 Savings Acc
 Current Acc
 int No. of days
 String Checkbook
}
class Savings Acc {
 int No. of days
 String Checkbook
}
class Current Acc {
 int No. of days
 String Checkbook
}

realme GT NEO 3T

```

import java.util.Scanner;

class Account {
    String name;
    String accountNumber;
    String type;
    double balance;
    Account(String name, String number, String type, double balance) {
        this.name = name;
        this.number = number;
        this.type = type;
        this.balance = balance;
    }
    void deposit(double amount) {
        if (amount > 0) {
            balance += amount;
            System.out.println("Deposit successful.");
            System.out.println("New balance: " + balance);
        } else {
            System.out.println("Invalid deposit amount.");
        }
    }
}

class CurrentAccount extends Account {
    final double MIN_BALANCE = 100;
    final double PAYOUT_Y = 100;
    CurrentAccount(String name, String number, double balance) {
        super(name, number, "Current", balance);
    }
    void withdraw(double amount) {
        if (amount > balance) {
            System.out.println("Insufficient balance.");
        }
    }
}

```

```

balance -= amount;
if (balance < MIN_BALANCE) {
    System.out.println ("Minimum balance requirement
not met. Penalty applied.");
    balance -= PENALTY;
}
showBalance();
}

void writeCheck (double amount) {
    if (amount <= balance) {
        balance -= amount;
        System.out.println ("Check written for : " + amount);
    } else {
        System.out.println ("Insufficient balance to
write the check.");
    }
    showBalance();
}

```

```

class SavingsAccount extends Account {
    double interestRate = 0.04;
    SavingsAccount (String name, int number, double balance) {
        super (name, number, "Savings", balance);
    }
}
```

```

void addInterest () {
    double interest = balance * interestRate;
    balance += interest;
    System.out.println ("Interest added. New Balance : "
balance);
}

```

}

realme GT NEO 3T

```

void withdraw(double amount) {
    if (amount > balance) {
        System.out.println("Insufficient balance.");
        return;
    }
    balance -= amount;
    showBalance();
}

public class Bank {
    public static void main(String[] args) {
        Scanner s = new Scanner(System.in);
        System.out.print("Enter account holder's name: ");
        String name = s.nextLine();
        System.out.print("Enter account number: ");
        int number = s.nextInt();
        System.out.print("Enter initial balance: ");
        double balance = s.nextDouble();
        System.out.print("Enter account type (1 for savings,
                        & 2 for current): ");
        int accountType = s.nextInt();

        Account account = (accountType == 1) ? new
        SavingsAccount(name, number, balance) : new
        CurrentAccount(name, number, balance);

        while (true) {
            System.out.println("\n---Action Menu---");
            System.out.println("1. Deposit 2. Withdraw
            3. Check Balance 4. Write check (current only) 5. Exit");
            int action = s.nextInt();
            if (action == 1) {
                System.out.print("Enter amount: ");
                double amount = s.nextDouble();
                account.deposit(amount);
            } else if (action == 2) {
                System.out.print("Enter amount: ");
                double amount = s.nextDouble();
                account.withdraw(amount);
            } else if (action == 3) {
                System.out.println("Current balance: " + account.getBalance());
            } else if (action == 4) {
                System.out.print("Enter amount: ");
                double amount = s.nextDouble();
                account.writeCheck(amount);
            } else if (action == 5) {
                break;
            }
        }
    }
}

```

realme GT NEO 3T

```

if (action == 5) {
    System.out.println ("fringe ..");
    s.close();
    return;
}

if (action == 1) {
    System.out.print ("Amount to deposit: ");
    account.deposit (s.nextDouble ());
}

else if (action == 2) {
    System.out.print ("Amount to withdraw: ");
    if (account instanceof SavingsAccount) {
        ((SavingsAccount) account).withdraw (s.nextDouble ());
    }
    else {
        ((CurrentAccount) account).withdraw (s.nextDouble ());
    }
}

else if (action == 3) {
    account.showBalance ();
}

else if (action == 4 & account instanceof CurrentAccount) {
    System.out.print ("Amount for check: ");
    ((CurrentAccount) account).writeCheck (s.nextDouble ());
}

else {
    System.out.println ("Invalid option. Try again.");
}

}

Output
Enter Account holder's name: Deepthi
Enter account number: 1BN22S137
Enter initial balance: 5000
Enter account type (1 for savings, 2 for current): 1

```

--Action Menu--

- 1. Deposit 2. Withdraw 3. Check Balance 4. Write check
(Current only) 5. Exit

Select an action: 1

Amount to deposit: 50000

Deposit successful. New balance: 505000.0

--Action Menu--

- 1. Deposit 2. Withdraw 3. Check Balance 4. Write check
(Current only) 5. Exit

Select an action: 2

Amount to withdraw: 504900

Current balance: 100.0

--Action Menu--

- 1. Deposit 2. Withdraw 3. Check Balance 4. Write check
(Current only) 5. Exit

Select an action: 5

Exiting...

Enter account holder's name: Keethi

Enter account number: 1BM23CS132

Enter initial balance: 5000

Enter account type (1 for Saving, 2 for Current): 2.

--Action Menu--

- 1. Deposit 2. Withdraw 3. Check Balance 4. Write check (Current only) 5. Exit

Select an action: 1

Amount to deposit: 10000

Deposit success realme GT NEO 3T : 405000.0

-- Action menu --
1. Deposit 2. withdraw 3. Check balance --
Select an action : 2
Amount to withdraw: 404900
Minimum balance requirement met. No penalty applied.
Current balance: 0.0

-- Action menu --
1. Deposit 2. Withdraw 3. Check Balance
Select an action : 4
Amount for check: 5000
Check ~~written~~ written for: 5000.0
Current balance: 195000.0

By
S2.10

realme GT NEO 3T

Code:

```
import java.util.Scanner;  
  
class Account {  
    String name;  
    String number;  
    String type;  
    double balance;  
  
    Account(String name, String number, String type, double balance) {  
        this.number = number;
```

```

        this.type = type;
        this.balance = balance;
    }

    void deposit(double amount) {
        if (amount > 0) {
            balance += amount;
            System.out.println("Deposit successful. New balance: " + balance);
        } else {
            System.out.println("Invalid deposit amount.");
        }
    }

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

class CurrentAccount extends Account {
    final double MIN_BALANCE = 1000;
    final double PENALTY = 100;

    CurrentAccount(String name, String number, double balance) {
        super(name, number, "Current", balance);
    }

    void withdraw(double amount) {
        if (amount > balance) {
            System.out.println("Insufficient balance.");
            return;
        }

        balance -= amount;
        if (balance < MIN_BALANCE) {
            System.out.println("Minimum balance requirement not met. Penalty applied.");
            balance -= PENALTY;
        }

        showBalance();
    }

    void writeCheck(double amount) {
        if (amount <= balance) {
            balance -= amount;
            System.out.println("Check written for: " + amount);
        } else {
            System.out.println("Insufficient balance to write the check.");
        }
    }
}

```

```

        }
    showBalance();
}
}

class SavingsAccount extends Account {
    double interestRate = 0.04;

    SavingsAccount(String name, String number, double balance) {
        super(name, number, "Savings", balance);
    }

    void addInterest() {
        double interest = balance * interestRate;
        balance += interest;
        System.out.println("Interest added. New balance: " + balance);
    }

    void withdraw(double amount) {
        if (amount > balance) {
            System.out.println("Insufficient balance.");
            return;
        }

        balance -= amount;
        showBalance();
    }
}

public class Bank {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter account holder's name: ");
        String name = scanner.nextLine();
        System.out.print("Enter account number: ");
        String number = scanner.nextLine();
        System.out.print("Enter initial balance: ");
        double balance = scanner.nextDouble();
        System.out.print("Enter account type (1 for Savings, 2 for Current): ");
        int accountType = scanner.nextInt();

        Account account = (accountType == 1) ? new SavingsAccount(name, number, balance) : new
CurrentAccount(name, number, balance);

        while (true) {
            System.out.println("\n--- Action Menu ---");

```

```

System.out.println("1. Deposit 2. Withdraw 3. Check Balance 4. Write Check (Current only)
5. Exit");
System.out.print("Select an action: ");
int action = scanner.nextInt();

if (action == 5) {
    System.out.println("Exiting...");
    scanner.close();
    return;
}

if (action == 1) {
    System.out.print("Amount to deposit: ");
    account.deposit(scanner.nextDouble());
} else if (action == 2) {
    System.out.print("Amount to withdraw: ");
    if (account instanceof SavingsAccount) {
        ((SavingsAccount) account).withdraw(scanner.nextDouble());
    } else {
        ((CurrentAccount) account).withdraw(scanner.nextDouble());
    }
} else if (action == 3) {
    account.showBalance();
} else if (action == 4 && account instanceof CurrentAccount) {
    System.out.print("Amount for check: ");
    ((CurrentAccount) account).writeCheck(scanner.nextDouble());
} else {
    System.out.println("Invalid action. Try again.");
}
}
}
}
Output:

```

```
D:\1BM23CS137>java Bank
Enter account holder's name: K.Keerthi Reddy
Enter account number: 1BM23CS137
Enter initial balance: 5000
Enter account type (1 for Savings, 2 for Current): 1

--- Action Menu ---
1. Deposit 2. Withdraw 3. Check Balance 4. Write Check (Current only) 5. Exit
Select an action: 1
Amount to deposit: 500000
Deposit successful. New balance: 505000.0

--- Action Menu ---
1. Deposit 2. Withdraw 3. Check Balance 4. Write Check (Current only) 5. Exit
Select an action: 2
Amount to withdraw: 504900
Current balance: 100.0

--- Action Menu ---
1. Deposit 2. Withdraw 3. Check Balance 4. Write Check (Current only) 5. Exit
Select an action: 5
Exiting...

D:\1BM23CS137>javac Bank.java

D:\1BM23CS137>java Bank
Enter account holder's name: K.Keerthi Reddy
Enter account number: 1BM23CS137
Enter initial balance: 5000
Enter account type (1 for Savings, 2 for Current): 2

--- Action Menu ---
1. Deposit 2. Withdraw 3. Check Balance 4. Write Check (Current only) 5. Exit
Select an action: 1
Amount to deposit: 400000
Deposit successful. New balance: 405000.0

--- Action Menu ---
1. Deposit 2. Withdraw 3. Check Balance 4. Write Check (Current only) 5. Exit
Select an action: 2
Amount to withdraw: 404900
Minimum balance requirement not met. Penalty applied.
Current balance: 0.0

--- Action Menu ---
1. Deposit 2. Withdraw 3. Check Balance 4. Write Check (Current only) 5. Exit
Select an action: 5
Exiting...
```

Program 6

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

Algorithm:

11/11/24 Lab Program-6 Bafna Gold Date: Page: 8

a) Create a package CIE which has two classes - Student and Internals. The class Student has members like usn, name, sem. The class Internals derived from Student has an array that stores the internal marks scored in 5 courses of the current semester of student. Create another package SEE which has the class External which is derived class of Student. This class has an array that stores SEE marks scored in 5 courses of the current semester of student. Import two packages in file that declares the final marks of n students in all 5 courses again.

```
* In CIE folder... CIE folder Student.java
package CIE;
import java.util.Scanner;
public class Student {
    protected String usn;
    protected String name;
    protected int sem;
    public void inputStudentDetails() {
        Scanner S = new Scanner(System.in);
        System.out.println("Enter USN : ");
        usn = S.nextLine();
        System.out.println("Enter Name : ");
        name = S.nextLine();
        System.out.println("Enter Semester : ");
        sem = S.nextInt();
    }
}
```

realme GT NEO 3T

```

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

/*
 * (show in Internal file)
 */
package CTET;
import java.util.Scanner;
public class Internal extends Student {
    protected int[] marks = new int[5];

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

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

/*
 * Now in SFC file External.java need to use *
 */
package SFC;
import CTET.Internal;
import java.util.Scanner;
realme GT NEO 3T

```

```

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

public void inputSECMarks() {
    Scanner s = new Scanner(System.in);
    System.out.println("Enter external Marks for Sec:");
    for (int i=0; i<5; i++) {
        System.out.print("Enter marks " + (i+1) + ":");
        externalMarks[i] = s.nextInt();
    }
}

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

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

```

realme GT NEO 3T

```

import java.util.*;
import java.util.Scanner;

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

        External[] students = new External[n];

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

        for (int i=0; i<n; i++) {
            students[i].calculateFinalMarks();
            students[i].displayFinalMarks();
        }

        sc.close();
    }
}

Output:
Enter number of students: 2
Enter details of student 1
realme GT NEO 3T

```

Enter USN:
137
Enter Name:
Kashvi
Enter Semester:
3
Enter Internal Marks for S courses:
Enter marks 1:
90
Enter marks 2:
90
Enter marks 3:
90
Enter marks 4:
90
Enter marks 5:
90
Enter External Marks for S subjects:
Enter marks 1:
90
Enter marks 2:
90
Enter marks 3:
90
Enter marks 4:
90
Enter marks 5:
90
Enter details for student 2
Enter USN:
142 realme GT NEO 3T
Enter name:

Kashvi
Enter semester:
3
Enter Internal Marks for S courses:
Enter marks 1:
85
Enter marks 2:
85
Enter marks 3:
85
Enter marks 4:
85
Enter marks 5:
85
Enter External Marks for Scusses:
Enter marks 1:
85
Enter marks 2:
85
Enter marks 3:
85
Enter marks 4:
85
Enter marks 5:
85
realme GT NEO 3T

Code:

/In CIE folder Internals.java/

```
package CIE;
```

```
import java.util.Scanner;
```

```
public class Internals extends Student {
```

```
    protected int[] marks = new int[5];
```

```
    public void inputCIEmarks() {
```

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

```
        System.out.println("Enter Internal Marks for 5 Courses: ");
```

```
        for (int i = 0; i < 5; i++) {
```

```
            System.out.println("Enter marks for course " + (i + 1) + ": ");
```

```
            marks[i] = s.nextInt();
```

```
        }
```

```
}
```

```
    public void displayCIEmarks() {
```

```
        System.out.println("Internal Marks: ");
```

```
        for (int i = 0; i < 5; i++) {
```

```
            System.out.println("Course " + (i + 1) + ": " + marks[i]);
```

```
        }
```

```
}
```

```
}
```

/In CIE folder Student.java/

```
package CIE;
```

```
import java.util.Scanner;
```

```
public class Student {
```

```
    protected String usn;
```

```
    protected String name;
```

```
    protected int sem;
```

```
    public void inputStudentDetails() {
```

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

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

```
        usn = s.nextLine();
```

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

```
        name = s.nextLine();
```

```
        System.out.println("Enter Semester: ");
```

```
        sem = s.nextInt();
```

```
}
```

```
    public void displayStudentDetails() {
```

```

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

```

/Now in SEE folder Externals.java needs to be saved/

```
package SEE;
```

```

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

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

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

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

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

```

/Now entering the main /

```
import SEE.External;
import java.util.Scanner;
public class Main {
```

```

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

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

```

Output:

```

Enter details for student 1
Enter USN:
137
Enter Name:
keerthi
Enter Semester:
3
Enter Internal Marks for 5 Courses:
Enter marks for course 1:
90
Enter marks for course 2:
90
Enter marks for course 3:
90
Enter marks for course 4:
90
Enter marks for course 5:
90
Enter External Marks for 5 Courses:
Enter marks for course 1:
90
Enter marks for course 2:
90
Enter marks for course 3:
90
Enter marks for course 4:
90
Enter marks for course 5:
90
Enter details for student 2
Enter USN:
142
Enter Name:
kachi
Enter Semester:
3
Enter Internal Marks for 5 Courses:
Enter marks for course 1:
95
Enter marks for course 2:
95
Enter marks for course 3:
95
Enter marks for course 4:
95
Enter marks for course 5:
95
Enter External Marks for 5 Courses:
Enter marks for course 1:
95
Enter marks for course 2:

```

```

File Edit View

package CIE;

import java.util.Scanner;

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

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

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

External.java X + File Edit View

package SEE;

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

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

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

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

```

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.

Algorithm:

28/11/21

Lab Program - 7

1) 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("Age Error");
    }
    public WrongAge(String message) {
        super(message);
    }
}
class InputScanner {
    Scanner s = new Scanner(System.in);
}
class Father extends InputScanner {
    int fatherAge;
    public Father() throws WrongAge {
        System.out.print("Enter father's age: ");
        fatherAge = s.nextInt();
        if (fatherAge < 0) {
            throw new Wrong("Age cannot be negative");
        }
    }
}
```

realme GT NEO 3T

```

public void display() {
    System.out.println("father's age : " + fatherAge);
}

class Son extends Father {
    int sonAge;
    public Son() throws WrongAge {
        super();
        System.out.print("Enter son's age : ");
        SonAge = s.nextInt();
        if (SonAge < 0) {
            throw new WrongAge ("Age cannot be negative");
        } else if (SonAge > fatherAge) {
            throw new WrongAge ("Son's age cannot be greater than father's age");
        }
    }

    display();
}

public class Fatherson {
    public static void main(String[] args) {
        try {
            Son son = new Son();
            son.display();
        } catch (WrongAge e) {
            System.out.println("Error : " + e.getMessage());
        }
    }
}

```

realme GT NEO 3T

```

System.out.println("USN : 1B23CS137")

```

Output:

```

Enter father's age : 50
Enter son's age : 18
Son's age : 18
Father's age : 50
Name : K. Keerthi Reddy
USN : 1B23CS137.

Enter father's age : 18
Enter son's age : 29
Son's age cannot be greater than or equal to father's age.
Name : K. Keerthi Reddy
USN : 1B23CS137

```

realme GT NEO 3T

Code:

```
import java.util.Scanner;

class WrongAge extends Exception {
    public WrongAge() {
        super("Age Error");
    }

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

class InputScanner {
    Scanner s = new Scanner(System.in);
}

class Father extends InputScanner {
    int fatherAge;

    public Father() throws WrongAge {
        System.out.print("Enter father's age: ");
        fatherAge = s.nextInt();

        if (fatherAge < 0) {
            throw new WrongAge("Age cannot be negative");
        }
    }

    public void display() {
        System.out.println("Father's age: " + fatherAge);
    }
}

class Son extends Father {
    int sonAge;

    public Son() throws WrongAge {
        super();

        System.out.print("Enter son's age: ");
        sonAge = s.nextInt();

        if (sonAge < 0) {
            throw new WrongAge("Age cannot be negative");
        } else if (sonAge > fatherAge) {
            throw new WrongAge("Son's age cannot be greater than father's age");
        }
    }
}
```

```

        }
    }

    public void display() {
        System.out.println("Son's age: " + sonAge);
    }
}

public class Fatherson {
    public static void main(String[] args) {
        try {
            Son son = new Son();
            son.display();
        } catch (WrongAge e) {
            System.out.println("Error: " + e.getMessage());
        }
        System.out.println("Name: K.keerthi Reddy");
        System.out.println("USN: 1BM23CS137");
    }
}

```

Output:

```

Enter son's age: 34
Son's age: 34

D:\1BM23CS137>javac Fatherson.java

D:\1BM23CS137>java Fatherson
Enter father's age: 50
Enter son's age: 35
Son's age: 35
Name: K.keerthi Reddy
USN: 1BM23CS137

D:\1BM23CS137>javac Fatherson.java

D:\1BM23CS137>java Fatherson
Enter father's age: -54
Error: Age cannot be negative
Name: K.keerthi Reddy
USN: 1BM23CS137

D:\1BM23CS137>javac Fatherson.java

D:\1BM23CS137>java Fatherson
Enter father's age: 45
Enter son's age: 79
Error: Son's age cannot be greater than father's age
Name: K.keerthi Reddy
USN: 1BM23CS137

D:\1BM23CS137>

```

```

File Edit View
import java.util.Scanner;

class WrongAge extends Exception {
    public WrongAge() {
        super("Age Error");
    }

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

class InputScanner {
    Scanner s = new Scanner(System.in);
}

class Father extends InputScanner {
    int fatherAge;

    public Father() throws WrongAge {
        System.out.print("Enter father's age: ");
        fatherAge = s.nextInt();

        if (fatherAge < 0) {
            throw new WrongAge("Age cannot be negative");
        }
    }

    public void display() {
        System.out.println("Father's age: " + fatherAge);
    }
}

class Son extends Father {
    int sonAge;

    public Son() throws WrongAge {
        super();
    }
}

```

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.

Algorithm:

28/11/12

Lab Program - 8

Q) 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 DisplayMessage1 extends Thread {

 public void run() {

 for (int i=0; i<5; i++) {

 try {

 System.out.println("BMS College of Engineering");

 Thread.sleep(2000);

 } catch (InterruptedException e) {

 System.out.println(e);

 }

 }

 }

}

class DisplayMessage2 extends Thread {

 public void run() {

 for (int i=0; i<5; i++) {

 try {

 System.out.println("CSE");

 Thread.sleep(2000);

 } catch (InterruptedException e) {

 System.out.println(e);

 }

 }

 }

}

public class multithread {

 public static void main (String [] args) {

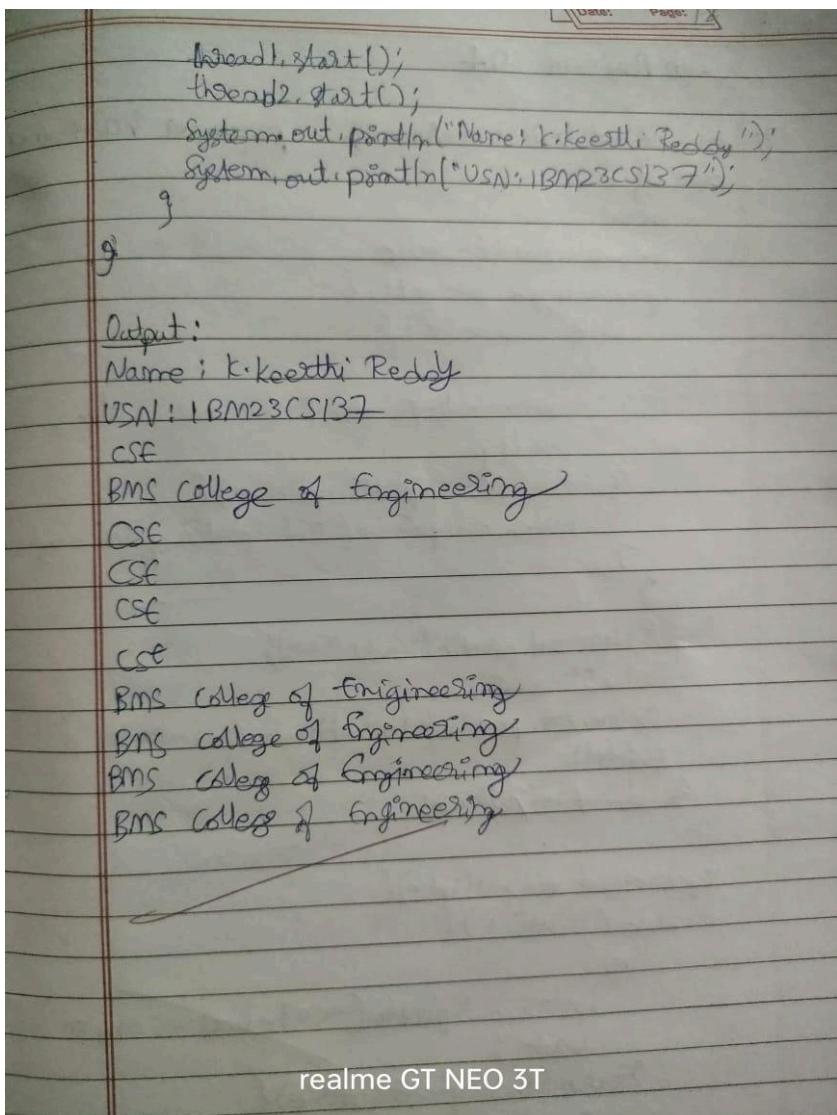
 DisplayMessage1 thread1 = new DisplayMessage1();

 DisplayMessage2 thread2 = new DisplayMessage2();

 thread1.start();

 thread2.start();

 }



Code:

```
class DisplayMessage1 extends Thread {
    public void run() {
        for(int i=0;i<5;i++) {
            try {
                System.out.println("BMS College of Engineering");
                Thread.sleep(10000);
            } catch (InterruptedException e) {
                System.out.println(e);
            }
        }
    }
}
```

```
class DisplayMessage2 extends Thread {
    public void run() {
        for(int i=0;i<5;i++) {
```

```

        try {
            System.out.println("CSE");
            Thread.sleep(2000);
        } catch (InterruptedException e) {
            System.out.println(e);
        }
    }
}

public class multithread{
    public static void main(String[] args) {
        DisplayMessage1 thread1 = new DisplayMessage1();
        DisplayMessage2 thread2 = new DisplayMessage2();

        thread1.start();
        thread2.start();
        System.out.println("Name: K.Keerthi Reddy");
        System.out.println("USN: 1BM23CS137");
    }
}

```

Output:

```

CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
^C
D:\1BM23CS137>javac multithread.java

D:\1BM23CS137>java multithread
Name: K.Keerthi Reddy
USN: 1BM23CS137
CSE
BMS College of Engineering
CSE
CSE
CSE
BMS College of Engineering
BMS College of Engineering
BMS College of Engineering
BMS College of Engineering

D:\1BM23CS137>

```

```

File Edit View

class DisplayMessage1 extends Thread {
    public void run() {
        for(int i=0;i<5;i++) {
            try {
                System.out.println("BMS College of Engineering");
                Thread.sleep(10000);
            } catch (InterruptedException e) {
                System.out.println(e);
            }
        }
    }
}

class DisplayMessage2 extends Thread {
    public void run() {
        for(int i=0;i<5;i++) {
            try {
                System.out.println("CSE");
                Thread.sleep(2000);
            } catch (InterruptedException e) {
                System.out.println(e);
            }
        }
    }
}

public class multithread{
    public static void main(String[] args) {
        DisplayMessage1 thread1 = new DisplayMessage1();
        DisplayMessage2 thread2 = new DisplayMessage2();

        thread1.start();
        thread2.start();
        System.out.println("Name: K.Keerthi Reddy");
        System.out.println("USN: 1BM23CS137");
    }
}

```

Program 9

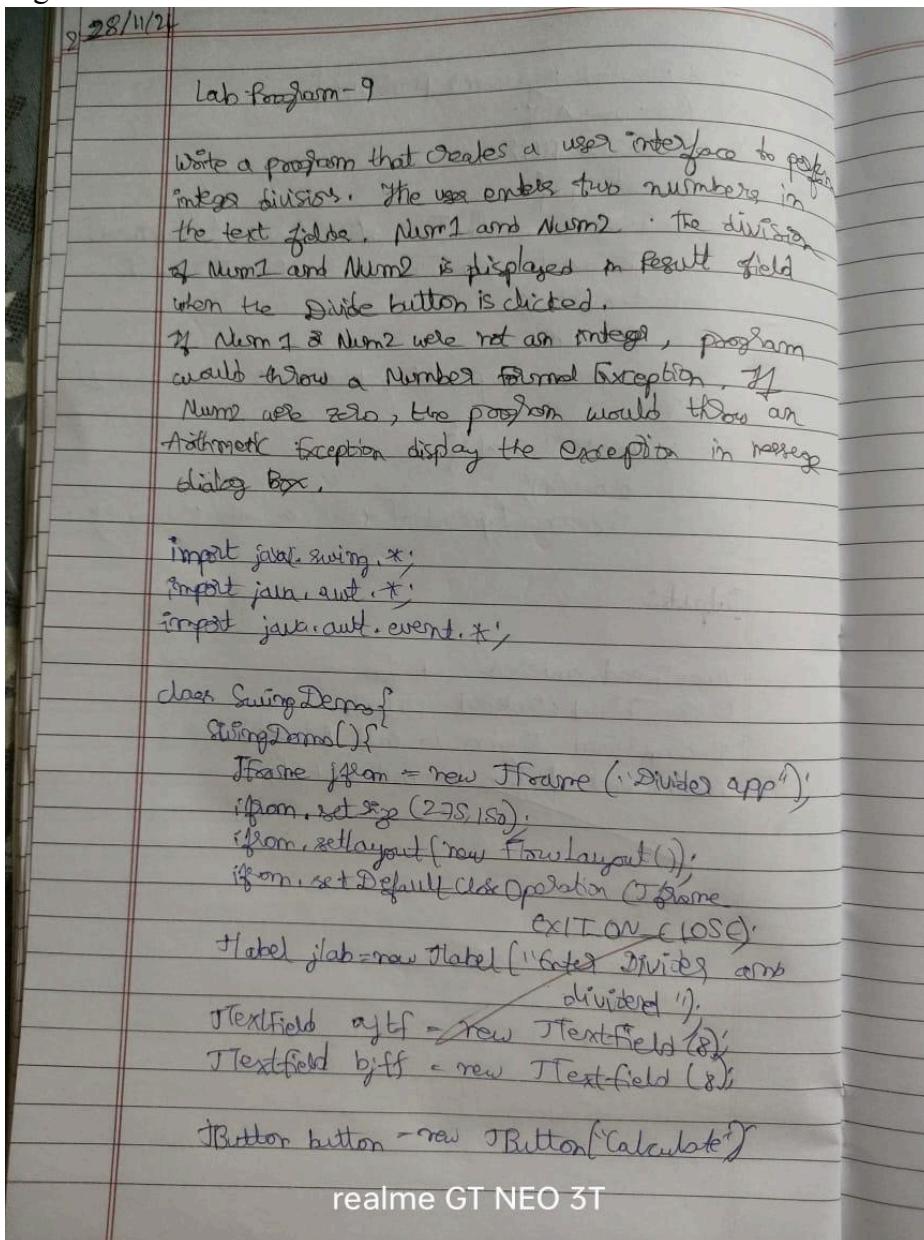
Write a program that creates a user interface to perform integer divisions.

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.

Algorithm:



```

JLabel err = new JLabel();
JLabel abt = new JLabel();
JLabel blab = new JLabel();
JLabel anslab = new JLabel();
if (from.add (err));
if (from.add (jlab));
if (from.add (abt));
if (from.add (blab));
if (from.add (anslab));

```

```

button.addActionListener (new ActionListener () {
    public void actionPerformed (ActionEvent evt) {
        try {
            int a = Integer.parseInt (tf1.getText ());
            int b = Integer.parseInt (tf2.getText ());
            int ans = a / b;
            alab.setText ("A = " + a);
            blab.setText ("B = " + b);
            anslab.setText ("Ans = " + ans);
            err.setText ("");
        } catch (NumberFormatException e) {
            alab.setText ("");
            blab.setText ("");
            anslab.setText ("");
            err.setText ("Enter only Integers");
        }
    }
});

```

realme GT NEO 3T

```

81
    Catch (ArithmetricException e) {
        alab.setText(" ");
        blab.setText(" ");
        anelab.setText(" ");
        erg.setText("B should be non zero! ");
    }
}

if (frm.isVisible(true)) {
    public static void main (String args[]) {
        SwingUtilities.invokeLater(new Runnable () {
            public void run () {
                new SwingDemo ();
            }
        });
    }
}

Output:

Divide App - □ X
Enter the divider and dividend:
52      14
Calculate A=52 B=14 A/B=3
realme GT NEO 3T

```

Code:

```

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

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

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

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

```

```

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

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

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

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

// Button action listener
button.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent evt) {
        try {
            int a = Integer.parseInt(ajtf.getText());
            int b = Integer.parseInt(bjtf.getText());
            int ans = a / b;

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

public static void main(String args[]) {

```

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

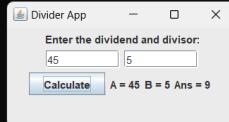
Output:

```
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
'C
D:\1BM23CS137>javac multithread.java

D:\1BM23CS137>java multithread
Name: K.Keerthi Reddy
JSN: 1BM23CS137
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
BMS College of Engineering
BMS College of Engineering
BMS College of Engineering

D:\1BM23CS137>javac SwingDemo.java

D:\1BM23CS137>java SwingDemo
```



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

class SwingDemo {
    SwingDemo() {
        JFrame jfrm = new JFrame("Divider App");
        jfrm.setSize(275, 150);
        jfrm.setLayout(new FlowLayout());
        jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        JLabel jlab = new JLabel("Enter the dividend and divisor:");
        JTextField ajtf = new JTextField(8);
        JTextField bjtf = new JTextField(8);
        JButton button = new JButton("Calculate");
        JLabel err = new JLabel();
        JLabel alab = new JLabel();
        JLabel blab = new JLabel();
        JLabel anslab = new JLabel();
        jfrm.add(jlab);
        jfrm.add(ajtf);
        jfrm.add(bjtf);
        jfrm.add(button);
        jfrm.add(alab);
        jfrm.add(blab);
        jfrm.add(anslab);
        jfrm.add(err);
        button.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent evt) {
                try {
                    int a = Integer.parseInt(ajtf.getText());
                    int b = Integer.parseInt(bjtf.getText());
                    int ans = a / b;
                    alab.setText("A = " + a);
                    blab.setText("B = " + b);
                    anslab.setText("Ans = " + ans);
                    err.setText("");
                } catch (NumberFormatException e) {
                    alab.setText("");
                    blab.setText("");
                    anslab.setText("");
                    err.setText("Please enter valid numbers");
                }
            }
        });
    }
}
```

Program 10a

An incorrect implementation of a producer and consumer using Inter process Communication.

Algorithm:

28/11/22
Lab Program 10A

Q) An incorrect implementation of a producer & consumer.

```
class Q {
    int n;
    boolean valueSet = false;
    synchronized int get() {
        while (!valueSet) {
            try {
                System.out.println("Consumer waiting");
                wait();
            } catch (InterruptedException e) {
                System.out.println("InterruptedException caught");
            }
        }
        System.out.println("Get: " + n);
        valueSet = false;
        System.out.println("Notify Producer");
        notify();
    }
    synchronized void put(int n) {
        while (valueSet) {
            try {
                System.out.println("Producer waiting");
                wait();
            } catch (InterruptedException e) {
                System.out.println("InterruptedException caught");
            }
        }
        this.n = n;
        valueSet = true;
    }
}
```

realme GT NEO 3T

System.out.println("Put: " + n)
System.out.println("In total items consumed: " + n)
notify();

```
}  
class Producer implements Runnable {  
    Queue q;  
    producer(Queue q) {  
        this.q = q;  
        new Thread(this, "Producer").start();  
    }  
    public void run() {  
        int i = 0;  
        while (i < 10) {  
            q.put(i++);  
        }  
    }  
}
```

```
}  
class Consumer implements Runnable {  
    Queue q;  
    consumer(Queue q) {  
        this.q = q;  
        new Thread(this, "Consumer").start();  
    }  
    public void run() {  
        int i = 0;  
        while (i < 10) {  
            int x = q.get();  
            System.out.println("Consumed: " + x);  
        }  
    }  
}
```

realme GT NEO 3T

```

2
close PCfixed {
    public static void main( String args[] ) {
        a -> rev Q()
        new Products( q );
        new Consumer( q );
        System.out.println( "Press control-c to stop" );
    }
}

3
Output:
Press control C to stop.
put: 0
Producer waiting...
Get: 0
put: 1
Producer waiting...
Consumed: 0
Get: 1
Consumed: 1
put: 2
Producer waiting...
Get: 2
Consumed: 2
put: 3
Producer waiting...
Get: 3
Consumed: 3
put: 4
Producer Waiting...
Get: 4
Consumed: 4
realme GT NEO 3T

```

Bafna Gold

```

5
Put: 5
Producer waiting...
Get: 5
Put: 6
Producer waiting...
Consumed: 5
Get: 6
Consumed: 6
Put: 7
Producer waiting...
Get: 7
Consumed: 7
Put: 8
Producer waiting...
Get: 8
Consumed: 8
Put: 9
Producer waiting...
Get: 9
Consumed: 9
Put: 10
Producer waiting...
Get: 10
Consumed: 10
Put: 11
Producer waiting...
Get: 11
Consumed: 11
Put: 12
Producer waiting...
Get: 11
Consumed: 11
realme GT NEO 3T

```

Code:

```
class Q {  
  
    int n;  
    boolean valueSet = false;  
    synchronized int get() {  
        while (!valueSet) {  
            try {  
                System.out.println("\nConsumer waiting\n");  
                wait();  
            } catch (InterruptedException e) {  
                System.out.println("InterruptedException caught");  
            }  
        }  
        System.out.println("Got: " + n);  
        valueSet = false;  
        System.out.println("\nIntimate Producer\n");  
        notify();  
        return n;  
    }  
    synchronized void put(int n) {  
        while (valueSet) {  
            try {  
                System.out.println("\nProducer waiting\n");  
                wait();  
            } catch (InterruptedException e) {  
                System.out.println("InterruptedException caught");  
            }  
        }  
        this.n = n;  
        valueSet = true;  
        System.out.println("Put: " + n);  
        System.out.println("\nIntimate Consumer\n");  
        notify();  
    }  
}  
  
class Producer implements Runnable {  
    Q q;  
    Producer(Q q) {  
        this.q = q;  
        new Thread(this, "Producer").start();  
    }  
  
    public void run() {  
        int i = 0;
```

```

        while (i < 15) {
            q.put(i++);
        }
    }
}

class Consumer implements Runnable {
    Q q;
    Consumer(Q q) {
        this.q = q;
        new Thread(this, "Consumer").start();
    }
    public void run() {
        int i = 0;
        while (i < 15) {
            int r = q.get();
            System.out.println("Consumed: " + r);
            i++;
        }
    }
}

class PCFixed {
    public static void main(String args[]) {
        Q q = new Q();
        new Producer(q);
        new Consumer(q);
        System.out.println("Press Control-C to stop.");
    }
}

```

Output:

Got: 12
Intimate Producer
Consumed: 12
Put: 13
Intimate Consumer
Producer waiting
Got: 13
Intimate Producer
Put: 14
Intimate Consumer
Consumed: 13
Got: 14
Intimate Producer
Consumed: 14
D:\IBM23CS137>javac Deadlock.java

D:\IBM23CS137>java Deadlock
Deadlock example by K. Keerthi Reddy (USN: IBM23CS137)
RacingThread entered B.bar
MainThread entered A.foo
RacingThread trying to call A.last()
Inside A.last
Back in other thread
MainThread trying to call B.last()
Inside B.last
Back in main thread

```

File Edit View
class B {
    synchronized void bar(A a) {
        String name = Thread.currentThread().getName();
        System.out.println(name + " entered B.bar");
        try {
            Thread.sleep(1000);
        } catch (Exception e) {
            System.out.println("B Interrupted");
        }
        System.out.println(name + " trying to call A.last()");
        a.last();
    }

    void last() {
        System.out.println("Inside B.last");
    }
}
class Deadlock implements Runnable {
    A a = new A();
    B b = new B();
    Deadlock() {
        Thread.currentThread().setName("MainThread");
        Thread t = new Thread(this, "RacingThread");
        t.start();
        a.foo(b);
        System.out.println("Back in main thread");
    }

    public void run() {
        b.bar(a);
        System.out.println("Back in other thread");
    }

    public static void main(String args[]) {
        System.out.println("Deadlock example by K. Keerthi Reddy (USN: IBM23CS137)");
        new Deadlock();
    }
}

```

Ln 37 Col 19 1453 characters 100% Windows (CRLF) UTF-8

Program 10b

Using Deadlock.

Algorithm:

28/11/2022
Algorithm - 10b

```
class A{  
    synchronized void foo(B b){  
        String name = Thread.currentThread().getName();  
        System.out.println(name + " entered A.foo()");  
        try{  
            Thread.sleep(1000);  
        } catch (Exception e){  
            System.out.println("A interrupted");  
        }  
        System.out.println("A interrupted");  
        b.last();  
    }  
    void last(){  
        System.out.println("Inside A.last()");  
    }  
}  
  
class B{  
    synchronized void bar(A a){  
        String name = Thread.currentThread().getName();  
        System.out.println(name + " entered B.bar()");  
        try{  
            Thread.sleep(1000);  
        } catch (Exception e){  
            System.out.println("B interrupted");  
        }  
        System.out.println("Name trying to call A.last()");  
        a.last();  
    }  
    void last(){  
        System.out.println("Inside B.last()");  
    }  
}
```

realme GT NEO 3T

```

class Deadlock implements Runnable {
    A a = new A();
    B b = new B();
    Deadlock() {
        Thread.currentThread().setName("main thread");
        Thread t = new Thread(this, "Racing Thread");
        t.start();
        a.foo(b);
        System.out.println("Back in main thread");
    }
    public void run() {
        b.bar(a);
        System.out.println("Back in other thread");
    }
}

Outputs:
MainThread entered A.foo
RacingThread entered B.bar
RacingThread trying to call A.last()
MainThread trying to call B.last()

```

realme GT NEO 3T

Code:

```

class A {
    synchronized void foo(B b) {
        String name = Thread.currentThread().getName();
        System.out.println(name + " entered A.foo");
        try {
            Thread.sleep(1000);
        } catch (Exception e) {
            System.out.println("A Interrupted");
        }
        System.out.println(name + " trying to call B.last()");
        b.last();
    }
}

```

```

        }
    void last() {
        System.out.println("Inside A.last");
    }
}
class B {
    synchronized void bar(A a) {
        String name = Thread.currentThread().getName();
        System.out.println(name + " entered B.bar");
        try {
            Thread.sleep(1000);
        } catch (Exception e) {
            System.out.println("B Interrupted");
        }

        System.out.println(name + " trying to call A.last()");
        a.last();
    }

    void last() {
        System.out.println("Inside B.last");
    }
}
class Deadlock implements Runnable {
    A a = new A();
    B b = new B();
    Deadlock() {
        Thread.currentThread().setName("MainThread");
        Thread t = new Thread(this, "RacingThread");
        t.start();
        a.foo(b);
        System.out.println("Back in main thread");
    }

    public void run() {
        b.bar(a);
        System.out.println("Back in other thread");
    }

    public static void main(String args[]) {
        System.out.println("Deadlock example by K. Keerthi Reddy (USN: 1BM23CS137)");
        new Deadlock();
    }
}

```

Output:

```
Got: 12
Intimate Producer
Consumed: 12
Put: 13
Intimate Consumer

Producer waiting
Got: 13
Intimate Producer
Put: 14
Intimate Consumer
Consumed: 13
Got: 14
Intimate Producer
Consumed: 14
D:\1BM23CS137>javac Deadlock.java

D:\1BM23CS137>java Deadlock
Deadlock example by K. Keerthi Reddy (USN: 1BM23CS137)
RacingThread entered B.bar
MainThread entered A.foo
RacingThread trying to call A.last()
Inside A.last
Back in other thread
MainThread trying to call B.last()
Inside B.last
Back in main thread
D:\1BM23CS137>

File Edit View
class B {
    synchronized void bar(A a) {
        String name = Thread.currentThread().getName();
        System.out.println(name + " entered B.bar");
        try {
            Thread.sleep(1000);
        } catch (Exception e) {
            System.out.println("B Interrupted");
        }
        System.out.println(name + " trying to call A.last()");
        a.last();
    }
    void last() {
        System.out.println("Inside B.last");
    }
}
class Deadlock implements Runnable {
    A a = new A();
    B b = new B();
    Deadlock() {
        Thread.currentThread().setName("MainThread");
        Thread t = new Thread(this, "RacingThread");
        t.start();
        a.foo(b);
        System.out.println("Back in main thread");
    }
    public void run() {
        b.bar(a);
        System.out.println("Back in other thread");
    }
    public static void main(String args[]) {
        System.out.println("Deadlock example by K. Keerthi Reddy (USN: 1BM23CS137)");
        new Deadlock();
    }
}

Ln 37, Col 19 1,453 characters 100% Windows (CRLF) UTF-8
```

