

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



**LAB REPORT**  
**on**  
**Object Oriented Java Programming**  
**(23CS3PCOOJ)**

*Submitted by*

Anag Kamlesh Devnani (1BM23CS031)

*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 **Anag Devnani (1BM23CS031)**, who is a bonafide student of **B.M.S. College of Engineering**. It is in partial fulfillment for the award of **Bachelor of Engineering in Computer Science and Engineering** of the Visvesvaraya Technological University, Belgaum. The Lab report has been approved as it satisfies the academic requirements in respect of an Object Oriented Java Programming (23CS3PCOOJ) work prescribed for the said degree.

Basavaraj Jakkali Assistant Professor Department of CSE, BMSCE	Dr. Jyothi S Nayak Professor & HOD Department of CSE, BMSCE
--	---

## Index

<b>Sl. No.</b>	<b>Date</b>	<b>Experiment Title</b>	<b>Page No.</b>
1	9/10	Quadratic Equation	4
2	16/10	SGPA Calculator	7
3	16/10	Class Book	11
4	23/10	Abstract Class Shape	15
5	30/10	Bank Class	20
6	13/11	Packages CIE and SEE	25
7	20/11	Exceptions in Inheritance Tree	30
8	27/11	Threads	33
9	27/11	IPC and Deadlock	35
10	27/11	User Interface for integer divisions	37

Github Link:

<https://github.com/AnagDevnani/BMSJavaLab/>

# Program 1 - Quadratic Equation

Algorithm:

equation  $ax^2 + bx + c$ . Read in  $a, b$  &  $c$  and use the quadratic formula. If the discriminant is  $b^2 - 4ac$  is negative, display a message stating that there are no real solutions.

A) import java.util.Scanner;  
import java.lang.Math;  
25 class Quadratic {  
 public static void main(String args[]) {  
 int a,b,c;  
 var con = System.console();  
 Scanner sc = new Scanner(con.reader());  
 System.out.print("Enter a: ");  
 30 a = sc.nextInt();  
 System.out.print("Enter b: ");

```

b = sc.nextInt();
System.out.print("Enter c: ");
c = sc.nextInt();
while(a==0) {
    System.out.println("\'a\' cannot be 0.");
    a = sc.nextInt();
}
double zero1, zero2;
double discriminant = b*b - 4*a*c;
if(discriminant > 0) {
    zero1 = (-b + Math.sqrt(discriminant))/(2*a);
    zero2 = (-b - Math.sqrt(discriminant))/(2*a);
    System.out.println("Real & Distinct Roots:");
    System.out.println("1st Root: " + zero1);
    System.out.println("2nd Root: " + zero2);
}
else if(discriminant == 0) {
    zero1 = -b/(2*a);
    System.out.println("Roots are real & equal");
    System.out.println("Roots: " + zero1);
}
else {
    System.out.println("No real roots");
}

```

O/P:

Enter a: 1	Enter a: 1	Enter a: 1
Enter b: 0	Enter b: 2	Enter b: 0
Enter c: -4	Enter c: 1	Enter c: 1
Real & Distinct Roots:	Roots & real & equal	No real roots
Root: 2.0 2nd Root: -2.0	Root: -1.0	

Code:

```

import java.util.Scanner;
import java.lang.Math;
class Quadratic{
    public static void main(String args[]){
        System.out.println("Program by Anag Devnani, USN: 1BM23CS031");
        int a,b,c;
        var con = System.console();
        Scanner sc = new Scanner(con.reader());
        System.out.print("Enter the value of a: ");
        a=sc.nextInt();
        System.out.print("Enter the value of b: ");
        b=sc.nextInt();
        System.out.print("Enter the value of c: ");
        c=sc.nextInt();
        while(a==0){
            System.out.println("\'a\' cannot be zero, please re-enter");
            a=sc.nextInt();
        }
        double zero1, zero2;
        double discriminant = (b*b-4*a*c);
        if(discriminant > 0){
            zero1=(-b+Math.sqrt(discriminant))/(2*a);
            zero2=(-b-Math.sqrt(discriminant))/(2*a);
            System.out.println("The roots are real and distinct:");
            System.out.println("1st Root: "+zero1);
            System.out.println("2nd Root: "+zero2);
        }
        else if(discriminant == 0){
            System.out.println("The roots are real and equal");
            zero1 = (-b/(2*a));
            System.out.println("Root: "+zero1);
        }
        else{
            System.out.println("The equation has no real roots");
        }
    }
}

```

```

[anag@archlinux Week 1]$ java Quadratic.java
Program by Anag Devnani, USN: 1BM23CS031
Enter the value of a: 1
Enter the value of b: 0
Enter the value of c: -4
The roots are real and distinct:
1st Root: 2.0
2nd Root: -2.0

```

```

[anag@archlinux Week 1]$ java Quadratic.java
Program by Anag Devnani, USN: 1BM23CS031
Enter the value of a: 1
Enter the value of b: 2
Enter the value of c: 1
The roots are real and equal
Root: -1.0

```

```

[anag@archlinux Week 1]$ java Quadratic.java
Program by Anag Devnani, USN: 1BM23CS031
Enter the value of a: 1
Enter the value of b: 0
Enter the value of c: 1
The equation has no real roots

```

## Program 2 - SGPA Calculator

### Week 2

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

D) import java.util.Scanner;

```
10 class Student {  
    String name = "Aneeq", usn = "031";  
    void display() {  
        System.out.println(name);  
        System.out.println(usn);  
    }  
}
```

```
15 double GPA() {  
    Scanner s = new Scanner(System.in);  
    System.out.println("Enter no. of subjects");  
    int n = s.nextInt();  
    int m[] = new int[n];  
    int c[] = new int[n];  
    int t_g = 0;  
    int t_c = 0;
```

```
20  
25 for (int i = 0; i < n; i++) {  
    System.out.print("Enter marks");  
    m[i] = s.nextInt();  
    c[i] = s.nextInt();  
}
```

```
30 if (int weight:  
    if ((m[i] < 100) && (m[i] >= 40)) {  
        weight = (m[i]/10) + 1;
```

else if ( $m[i] == 100$ ) {  
    weight = 10;  
    y

5     else{

    weight = 0;

    y

    t-g += weight + c[i];  
    t-c += c[i];

10     y

    return t-g / t-c;

    y

    y

15     class SGPA {

        public static void main (String [] args) {  
            Student anag = new Student();  
            System.out.println (anag.GPA());  
        y

20     y

O/P:

Enter no. of subjects = 2

25

Enter marks : 100

1

Enter marks : 0

30

1

Re  
1610124

5.0

Code:

```

import java.util.Scanner;

class Student{
    //basic student details
    String name="Anag", usn="031";
    void display(){
        System.out.println("Name: "+name);
        System.out.println("USN: "+usn);
    }

    double GPA(){
        //SGPA variable init
        Scanner s = new Scanner(System.in);
        System.out.print("Enter number of subjects: ");
        int no_of_subjects = s.nextInt();
        System.out.print("\n");
        int marks_array[] = new int[no_of_subjects];
        int credits_array[] = new int[no_of_subjects];
        int total_gradepoint = 0;
        int total_credits = 0;

        for(int i = 0; i < no_of_subjects; i++){
            //defining the arrays
            System.out.print("Enter the marks of subject "+(i+1)+": ");
            marks_array[i] = s.nextInt();
            System.out.print("Enter the credits of subject "+(i+1)+": ");
            credits_array[i] = s.nextInt();

            //performing the calculations
            int marks_weightage;
            if((marks_array[i]<100) && (marks_array[i]>=40)){
                marks_weightage = (marks_array[i]/10) + 1;
            }
            else if (marks_array[i]==100){
                marks_weightage = 10;
            }
            else{
                marks_weightage = 0;
            }
            total_gradepoint += marks_weightage * credits_array[i];
            total_credits += credits_array[i];
        }

        return total_gradepoint/total_credits;
    }
}

class SGPA{
    public static void main(String[] args){
        System.out.print("Anag Devnani\nIBM23CS031\n");
        Student anag = new Student();
        System.out.println("\n"+anag.GPA());
    }
}

```

```
bmscecse@bmscecse-HP-Elite-Tower-600-G9-Desktop-PC:~/Downloads$ javac SGPA.java
bmscecse@bmscecse-HP-Elite-Tower-600-G9-Desktop-PC:~/Downloads$ java SGPA
Anag Devnani
1BM23CS031
Enter number of subjects: 3

Enter the marks of subject 1: 10
Enter the credits of subject 1: 3
Enter the marks of subject 2: 50
Enter the credits of subject 2: 4
Enter the marks of subject 3: 80
Enter the credits of subject 3: 2

4.0
```

## Program 3 - Class Book

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 a book object.

```
import java.util.Scanner;  
class Book {  
    10    String name, author;  
    int num-pages;  
    double price;  
  
    void setDetails () {  
        15    Scanner sc = new Scanner (System.in);  
        System.out.print ("Enter name, author, pages, price");  
        sc.next();  
        this.name = sc.next();  
        this.author = sc.next();  
        this.num-pages = sc.nextInt();  
        20    this.price = sc.nextDouble();  
        return;  
    }  
  
    void getDetails () {  
        25    System.out.println ("Name: " + name + "\nAuthor: " + author +  
        "\nPages: " + num-pages + "\nPrice: " + price);  
        return;  
    }  
  
    public String toString () {  
        30    return "Name: " + name + "\nAuthor: " + author + "\nPages: "  
        num-pages + "\nPrice: " + price);  
    }  
}
```

```

class BookDemo {
    public static void main (String args [ ] ) {
        Scanner sc = new Scanner (System.in);
        System.out.print ("No. of Books: ");
        int bookNum = sc.nextInt();
        Book bookArray [] = new Book [bookNum];
        for (int i = 0; i < bookNum; i++) {
            bookArray[i] = new Book();
            bookArray[i].setDetails();
        }
    }
}

```

```

for (int i = 0, i < bookNum, i++) {
    bookArray[i].getDetails();
}

```

```

y
y
y

```

O/p:

Enter No. of Books : 2

Enter Name: Ram

Enter Author: Ramayan

Enter Pages: 200

Enter Price: 150

Enter Name: Cat in a Hat

Enter Author: Dr. Seuss

Enter Pages: 50

Enter Price: 100

Name: Ram

Author: Ramayan

Pages: 200

Price: 150

Name: Cat in a Hat

Author: Dr. Seuss

Pages: 50

Price: 100

P

23/01/24

```

import java.util.Scanner;
class Book{
    String name, author;
    int num_pages;
    double price;

    void setDetails(){
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter Name:");
        this.name = sc.next();
        System.out.print("Enter Author:");
        this.author = sc.next();
        System.out.print("Enter Pages:");
        this.num_pages = sc.nextInt();
        System.out.print("Enter Price:");
        this.price = sc.nextDouble();
        return;
    }

    void getDetails(){
        System.out.println("Name: "+name+"\nAuthor: "+author+"\nPages: "+num_pages+"\nPrice: "+price);
        return;
    }

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

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

        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the number of books: ");
        int bookNum = sc.nextInt();

        Book bookArray[] = new Book[bookNum];

        for(int i = 0; i<bookNum; i++){
            bookArray[i] = new Book();
            bookArray[i].setDetails();
            System.out.println();
        }

        for (int i = 0; i<bookNum; i++){
            bookArray[i].getDetails();
            System.out.println();
        }
    }
}

```

```
[anag@archlinux Week 3]$ java BookDemo.java
Enter the number of books: 3
Enter Name:Book1
Enter Author:Autho1
Enter Pages:1
Enter Price:1

Enter Name:Book1
Enter Author:Author2
Enter Pages:2
Enter Price:2

Enter Name:Book3
Enter Author:Authout3
Enter Pages:3
Enter Price:3

Name: Book1
Author: Autho1
Pages: 1
Price: 1.0

Name: Book1
Author: Author2
Pages: 2
Price: 2.0

Name: Book3
Author: Authout3
Pages: 3
Price: 3.0
```

## Program 4 - Abstract Class Shape

Date 23/10/2023

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 of the classes contain only the method printArea() that prints the area of the given shape.

```
import java.util.Scanner;
class InputScanner {
    Scanner sc = new Scanner(System.in);
}

abstract class Shape extends InputScanner {
    double dim1;
    double dim2;
    abstract double printArea();
}

class Rectangle extends Shape {
    Rectangle() {
        System.out.println("Enter dimension of rectangle");
        super.dim1 = sc.nextInt();
        super.dim2 = sc.nextInt();
    }

    double printArea() {
        System.out.println("\nArea of rectangle:");
        return (dim1 * dim2);
    }
}
```

class Triangle extends shape {

Triangle () {

System.out.println("Enter dimensions of Triangle: ");

super.dim1 = sc.nextInt();

super.dim2 = sc.nextInt();

y

double printArea () {

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

return (0.5 \* dim1 \* dim2);

y

class Circle extends shape {

Circle () {

System.out.println("Enter radius of circle: ");

super.dim1 = sc.nextInt();

y

double printArea () {

System.out.println("Area of Circle: ");

return (3.14 \* dim1 \* dim2);

y

class AbstractDemo {

public static void main (String args[]) {

Rectangle r = new Rectangle();

Triangle t = new Triangle();

Circle c = new Circle();

Shape figet;

FigRef = 8;

System.out.println("Area is: " + FigRef.printArea());

FigRef = f;

System.out.println("Area is: " + FigRef.printArea());

FigRef = c;

System.out.println("Area is: " + FigRef.printArea());

y

y

O/P :

Enter dimensions of Rectangle:

1 2

Enter dimensions of Triangle:

1 2

Enter radius of Circle:

1

Area of Rectangle:

2.0

Area of Triangle:

1.0

25 Area of Circle:

3.14

~~R<sub>2</sub>~~

~~23/10/24~~

30

```

import java.util.Scanner;
class InputScanner{
    Scanner sc = new Scanner(System.in);
}

abstract class Shape extends InputScanner{
    double dim1;
    double dim2;
    abstract double printArea();
}
class Rectangle extends Shape{

    Rectangle(){
        System.out.println("Enter the dimensions of the Rectangle: ");
        super.dim1 = sc.nextInt();
        super.dim2 = sc.nextInt();
    }

    double printArea(){
        System.out.println("\nArea of rectangle: ");
        return(dim1 * dim2);
    }
}

class Triangle extends Shape{

    Triangle (){
        System.out.println("Enter the dimensions of the Triangle: ");
        super.dim1 = sc.nextInt();
        super.dim2 = sc.nextInt();
    }

    double printArea(){
        System.out.println("Area of Triangle: ");
        return 0.5 * dim1 * dim2;
    }
}

class Circle extends Shape{
    Circle (){
        System.out.println("Enter the dimension (radius) of the Circle: ");
        super.dim1 = sc.nextInt();
    }

    double printArea(){
        System.out.println("Area of Circle: ");
        return 3.14*dim1*dim1;
    }
}

class AbstractDemo{
    public static void main(String args[]){
        Rectangle r = new Rectangle();
        Triangle t = new Triangle();
        Circle c = new Circle ();

        Shape figref; //This is OK, no object is created

        figref = r;
        System.out.println("Area is: "+figref.printArea()+"\n");

        figref = t;
        System.out.println("Area is: "+ figref.printArea()+"\n");

        figref = c;
        System.out.println("Area is: "+figref.printArea()+"\n");
    }
}

```

```
[anag@archlinux Week 4]$ java AbstractDemo
Enter the dimensions of the Rectangle:
1
2
Enter the dimensions of the Triangle:
1 2
Enter the dimension (radius) of the Circle:
1

Area of rectangle:
Area is: 2.0

Area of Triangle:
Area is: 1.0

Area of Circle:
Area is: 3.14
```

## Program 5 - Bank Class

```
3) #Bank
class Account {
    private String name;
    private String account;
    private double bal;
    public Account (String name, String accnum, double bal) {
        this.name = name;
        this.accnum = accnum;
        this.bal = bal;
    }
    public void deposit (double amt) {
        if (amt > 0) {
            bal += amt;
            System.out.println("Deposited " + amt);
        } else {
            System.out.println("Insufficient Funds!");
        }
    }
    public double getBal() {
        return bal;
    }
    public void setBal (double bal) {
        this.bal = bal;
    }
}
class SavAcct extends Account
```

```

private static final double int-rate = 0.05;
public SavAct(String name, String accnum, double bal) {
    super(name, accnum, bal);
}
public void addInterest() {
    double interest = getBal() * int-rate;
    setBal(getBal() + interest);
    System.out.println("Interest added: " + interest);
}
class CurAct extends Account {
    public static final double min-bal = 500;
    public static final double penalty = 50;
    public CurAct(String name, String accnum, double bal) {
        super(name, accnum, bal);
    }
    public void withdraw(double amt) {
        if (amt > 0 && getBal() - amt >= 0) {
            setBal(getBal() - amt);
            System.out.println("Withdraw: " + amt);
            checkMinBal();
        } else {
            System.out.println("Insufficient Funds");
        }
    }
    private void checkMinBal() {
        if (getBal() < min-bal) {
            setBal(getBal() - penalty);
            System.out.println("Below min bal, penalty: " + penalty);
        }
    }
}

```

public class Bank {

```

    public static void main(String[] args) {
        SavAcc savacc = new SavAcc("Rahul", "Rahul13", 100);
        CurrAcc curacc = new CurrAcc("Rohan", "Rohan14", 500);
        System.out.println("Savings Account:");
        savacc.deposit(500);
        savacc.displayBal();
        savacc.addInterest();
        savacc.withdraw(200);
        savacc.displayBal();
    }

```

System.out.println("Current Acc:");

curacc.deposit(1000);

curacc.displayBal();

curacc.withdraw(1000);

curacc.displayBal();

curacc.withdraw(200);

curacc.displayBal();

3

O/P:

Savings Account

Deposited: 500.0

Balance: 500.0

Interest Added: 25.0

Withdraw: 200.0

Balance: 300.0

Cur Acc:

Deposited: 1000.0

Balance: 1000.0

Withdraw: 1000.0

Balance: 1000.0

Withdraw: 200.0

25

30

For  
13/11/24

```

class Account{
private String name;
private String account;
private double bal;

public Account (String name, String accnum, double ball){
this.name = name;
this.account = account;
this.bal = bal;
}

public void deposit (double amt){
if (amt > 0){
bal+=amt;
System.out.println("Deposited" + amt);
}
else{
System.out.println("Invalid Deposit");
}
}

public void displayBal(){
System.out.println("Balance" + bal);
}

public void withdraw (double amt){
if (amt>0 && amt<=bal){
bal-=amt;
System.out.println("Withdraw: "+amt);
}
else{
System.out.println("Insufficient Funds");
}
}

public double getBal(){
return bal;
}

public void setBal(double Bal){
this.bal = bal;
}

class SavAct extends Account{
private static final double int_rate = 0.05;
public SavAct (String name, String accnum, double bal){
super(name, accnum, bal);
}

public void addInterest(){
double interest = getBal() * int_rate;
setBal(getBal() + interest);
System.out.println("Interest added:" + interest);
}

class CurAct extends Account{
public static final double min_bal = 500;
public static final double penalty = 50;
public CurAct(String name, String accnum, double bal){
super(name, accnum, bal);
}

public void withdraw (double amt){
if (amt>0 && getBal() - amt>=0){
setBal(getBal() - amt);
System.out.println("Withdraw: " + amt);
checkMinBal();
}
else{
System.out.println("Insufficient Funds");
}
}

private void checkMinBal(){
if (getBal() < min_bal){
setBal(getBal()-penalty);
System.out.println("Below min balance, penalty: "+penalty);
}
}
}

public class Bank{
public static void main(String[], args){
SavAct savacc = new SavAct("Rahul", "Rahul13",1000);
CurAct curacc = new CurAct("Rohan", "Rohan14",500);
System.out.println("Savings Account: ");
savacc.deposit(500);
savacc.displayBal();
savacc.addInterest();
savacc.withdraw(200);
savacc.displayBal();

System.out.println("\n Current Account: ");
curacc.deposit(1000);
curacc.displayBal();
curacc.withdraw(1000);
curacc.displayBal();
curacc.withdraw(200);
}
}

```

```
bmscecse@bmscecse-HP-Elite-Tower-600-G9-Desktop-PC:~/Downloads$ javac Bank.java
bmscecse@bmscecse-HP-Elite-Tower-600-G9-Desktop-PC:~/Downloads$ java Bank
Savings Account:
Deposit 500.0
Balance 500.0
Interest added: 25.0
Withdraw: 200.0
Balance 300.0

Current Account:
Deposit 1000.0
Balance 1000.0
Withdraw: 1000.0
Balance 1000.0
Withdraw: 200.0
```

## Program 6 - Packages (CIE & SEE)

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.

### 1. Student.java

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;

    public void displayStudentInfo () {

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

        System.out.println("Name : " + name);

        System.out.println("Semester : " + sem);

    }

### 2. Internals.java

package cie;

public class Internals extends Student {

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

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

        super(usn, name, sem);

        this.internalMarks = internalMarks;

    }

```
public void displayInternalMarks() {
    SOP("Internal marks : ");
    for (int mark : internalMarks) {
        SOP(mark + " ");
    }
}
```

### 3. External.java

```
package see;
```

```
import cie.Student;
```

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

```
public void displayExternalMarks() {
    SOP("Ext marks : ");
    for (int mark : externalMarks) {
        SOP(mark + " ");
    }
}
```

```
SOP();
```

O/P:

Student Info :

UIN: USPU123

Name: Anug

Semester: 3

Internal Marks: 20 30 25

External Marks: 60 70 80 6

Final Marks: 80 100 80 9

3/11/29

```

package see;

import cie.Student;

public class External extends Student {
    public int[] externalMarks = new int[5];

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

    public void displayExternalMarks() {
        System.out.println("External Marks: ");
        for (int mark : externalMarks) {
            System.out.print(mark + " ");
        }
        System.out.println();
    }
}

package cie;

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

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

    public void displayInternalMarks() {
        System.out.println("Internal Marks: ");
        for (int mark : internalMarks) {
            System.out.print(mark + " ");
        }
        System.out.println();
    }
}

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;
    }

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

```

```

import cie.Internals;
import see.External;

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

        int n = 2;

        int[] internalMarks1 = {20, 30, 25, 28, 22};
        int[] externalMarks1 = {60, 70, 55, 65, 50};
        Internals student1Internal = new Internals("USN123", "Alice", 3, internalMarks1);
        External student1External = new External("USN123", "Alice", 3, externalMarks1);

        int[] internalMarks2 = {18, 25, 20, 23, 28};
        int[] externalMarks2 = {50, 65, 60, 58, 45};
        Internals student2Internal = new Internals("USN124", "Bob", 3, internalMarks2);
        External student2External = new External("USN124", "Bob", 3, externalMarks2);

        System.out.println("Student 1 Info: ");
        student1Internal.displayStudentInfo();
        student1Internal.displayInternalMarks();
        student1External.displayExternalMarks();

        int[] finalMarks1 = calculateFinalMarks(student1Internal.internalMarks, student1External.externalMarks);
        displayFinalMarks(finalMarks1);

        System.out.println("\nStudent 2 Info: ");
        student2Internal.displayStudentInfo();
        student2Internal.displayInternalMarks();
        student2External.displayExternalMarks();

        int[] finalMarks2 = calculateFinalMarks(student2Internal.internalMarks, student2External.externalMarks);
        displayFinalMarks(finalMarks2);
    }

    public static int[] calculateFinalMarks(int[] internalMarks, int[] externalMarks) {
        int[] finalMarks = new int[5];
        for (int i = 0; i < 5; i++) {
            finalMarks[i] = internalMarks[i] + externalMarks[i];
        }
        return finalMarks;
    }

    public static void displayFinalMarks(int[] finalMarks) {
        System.out.println("Final Marks (Internal + External): ");
        for (int mark : finalMarks) {
            System.out.print(mark + " ");
        }
        System.out.println();
    }
}

```

```
bmscecse@bmscecse-HP-Elite-Tower-600-G9-Desktop-PC:~/Downloads$ cd 'Week 6'  
bmscecse@bmscecse-HP-Elite-Tower-600-G9-Desktop-PC:~/Downloads/Week 6$ javac -d ./Main.java  
error: no source files  
bmscecse@bmscecse-HP-Elite-Tower-600-G9-Desktop-PC:~/Downloads/Week 6$ javac Main.java  
bmscecse@bmscecse-HP-Elite-Tower-600-G9-Desktop-PC:~/Downloads/Week 6$ java Main  
Student 1 Info:  
USN: USN123  
Name: Alice  
Semester: 3  
Internal Marks:  
20 30 25 28 22  
External Marks:  
60 70 55 65 50  
Final Marks (Internal + External):  
80 100 80 93 72  
  
Student 2 Info:  
USN: USN124  
Name: Bob  
Semester: 3  
Internal Marks:  
18 25 20 23 28  
External Marks:  
50 65 60 58 45  
Final Marks (Internal + External):  
68 90 80 81 73
```

## Program 7 - Exceptions in Inheritance Tree

Week-7: Father Son Exception handling

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

class SonAgeException extends Exception {
    public SonAgeException (String message) {
        super(message);
    }
}

class Father {
    private int age;
    public Father (int age) throws WrongAgeException {
        if (age < 0) {
            throw new WrongAgeException ("Wrong Age");
        }
        this.age = age;
    }
    public int getAge () {
        return age;
    }
}

class Son extends Father {
    private int sonAge;
    public Son (int fatherAge, int sonAge) throws WrongAgeException, SonAgeException {
        super(fatherAge);
        this.sonAge = sonAge;
    }
}
```

```

if (sonAge >= fatherAge) {
    throw new SonAgeException ("Son's age cannot
    be greater than father's");
}

5.    if (sonAge == sonAge) {
        public int getSonAge () {
            return sonAge;
        }
    }

10.   public class FatherSon {
        public static void main (String [] args) {
            while (true) {
                Scanner sc = new Scanner (System. in);
                System.out.println ("Father's Age : ");
                int fatherage = sc.nextInt ();
                int sonage = sc.nextInt ();
                try {
                    Son son = new Son (fatherage, sonage);
                    System.out.println ("Accepted successfully");
                } catch (WrongAgeException e) {
                    System.out.println (e.getMessage ());
                } catch (SonAgeException e) {
                    System.out.println (e.getMessage ());
                }
            }
        }
    }
}

```

```

/*
Q) Write a program that demonstrates handling of exceptions in inheritance tree.
Create a base class called "Father" and a derived class called "Son" which extends the base class.
In Father's class implement a constructor which takes the age and throws the exception wrongAge() when the input age is less than zero.
In Son's class implement a constructor that uses father and son's age and throws an exception if son's age is greater than or equal to father's age.
*/
import java.util.Scanner;
class WrongAgeException extends Exception {
    public WrongAgeException(String message) {
        super(message);
    }
}

class SonAgeException extends Exception {
    public SonAgeException(String message) {
        super(message);
    }
}

class Father {
    private int age;
    public Father(int age) throws WrongAgeException {
        if (age < 0) {
            throw new WrongAgeException("Wrong age");
        }
        this.age = age;
    }
    public int getAge() {
        return age;
    }
}

class Son extends Father {
    private int sonAge;
    public Son(int fatherAge, int sonAge) throws WrongAgeException, SonAgeException {
        super(fatherAge);
        if (sonAge >= fatherAge) {
            throw new SonAgeException("Son's age cannot be greater than or equal to father's age");
        }
        this.sonAge = sonAge;
    }
    public int getSonAge() {
        return sonAge;
    }
}
public class FatherSon{
    public static void main(String[] args) {
        while(true){
            Scanner sc = new Scanner(System.in);
            System.out.print("Enter Father's Age: ");
            int fatherAge = sc.nextInt();
            System.out.print("Enter Son's Age: ");
            int sonAge = sc.nextInt();
            try {
                Son son = new Son(fatherAge, sonAge);
                System.out.println("Accepted Succesfully");
            }
            catch (WrongAgeException e) {
                System.out.println(e.getMessage());
            }
            catch (SonAgeException e) {
                System.out.println(e.getMessage());
            }
            System.out.println("Would you like to re-enter details (Y/n)");
            String input = sc.next();
            if (input.equalsIgnoreCase("n")) {
                break;
            }
        }
    }
}

```

```

bmscecse@bmscecse-HP-Elite-Tower-600-G9-Desktop-PC:~/Downloads$ javac FatherSon.java
bmscecse@bmscecse-HP-Elite-Tower-600-G9-Desktop-PC:~/Downloads$ java FatherSon
Enter Father's Age: 20
Enter Son's Age: 10
Accepted Succesfully
Would you like to re-enter details (Y/n)

Y
Enter Father's Age: 20
Enter Son's Age: 21
Son's age cannot be greater than or equal to father's age
Would you like to re-enter details (Y/n)

```

## Program 8 - Multithreading

Week 8: Multithreading

```
20 class BMS extends Thread {  
    public void run() {  
        try {  
            while (true) {  
                System.out.println("BMS College  
of Engineering");  
                Thread.sleep(10000);  
            }  
        } catch (InterruptedException e) {}  
    }  
  
25 class CSE extends Thread {  
    public void run() {  
        try {  
            while (true) {  
                System.out.println("CSE");  
                Thread.sleep(2000);  
            }  
        } catch (InterruptedException e) {}  
    }  
  
30 public class Multithreading {  
    public static void main (String [] args) {  
        BMS bms = new BMS();  
        CSE cse = new CSE();  
        bms.start();  
        cse.start();  
    }  
}
```

O/P:

CSE  
BMS College  
CSE  
CSE

```

class BMS extends Thread {
    public void run() {
        try {
            while (true) {
                System.out.println("BMS College of Engineering");
                Thread.sleep(10000); // Sleep for 10 seconds
            }
        }catch (InterruptedException e) {}
    }
}
class CSE extends Thread {
    public void run() {
        try {
            while (true) {
                System.out.println("CSE");
                Thread.sleep(2000); // Sleep for 2 seconds
            }
        }catch (InterruptedException e) {}
    }
}

public class Multithreading{
    public static void main(String[] args) {
        BMS bms = new BMS();
        CSE cse = new CSE();
        bms.start();
        cse.start();
    }
}

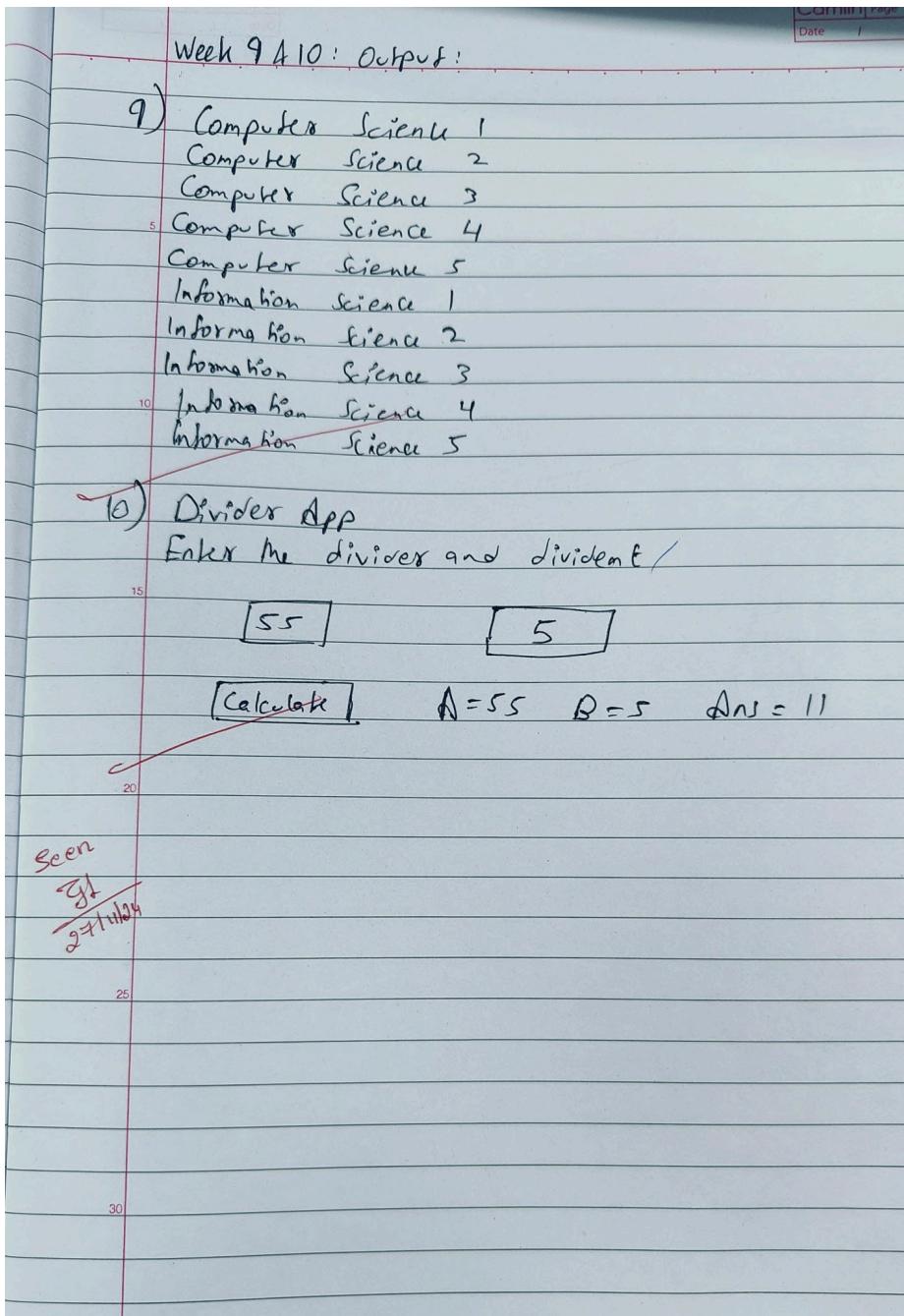
```

```

bmscecse@bmscecse-HP-Elite-Tower-600-G9-Desktop-PC:~/Downloads$ javac Multithreading.java
bmscecse@bmscecse-HP-Elite-Tower-600-G9-Desktop-PC:~/Downloads$ java Multithreading
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE

```

## Program 9 - IPC and Deadlock



```

class CS extends Thread
{
public void run()
{
for(int i=1; i<=5; i++)
System.out.println("Computer Science " + i);
}
}

class IS extends Thread
{
public void run()
{
for(int i=1; i<=5; i++)
System.out.println("Information Science " + i);
}
}

public class CS_IS_ThreadProgram {
public static void main(String args[])
{
CS c1 = new CS();
c1.start();

IS i1 = new IS();
i1.start();
}
}

```

```

bmscecse@bmscecse-HP-Elite-Tower-600-G9-Desktop-PC:~/Downloads$ javac CS_IS_ThreadProgram.java
bmscecse@bmscecse-HP-Elite-Tower-600-G9-Desktop-PC:~/Downloads$ java CS_IS_ThreadProgram
Information Science 1
Computer Science 1
Information Science 2
Computer Science 2
Information Science 3
Computer Science 3
Information Science 4
Computer Science 4
Information Science 5
Computer Science 5

```

## Program 10 - User interface for integer divisions

Week 9 & 10: Output:

9) Computer Science 1  
Computer Science 2  
Computer Science 3  
Computer Science 4  
Computer Science 5  
Information Science 1  
Information Science 2  
Information Science 3  
Information Science 4  
Information Science 5

10) Divider App  
Enter the divisor and dividend /

15      [55]      [5]

[Calculate]      A=55      B=5      Ans = 11

20

Scen  
gt  
27/11/24

25

30

```

import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
class SwingDemo{
SwingDemo(){
// create jframe container
JFrame jfrm = new JFrame("Divider App");
jfrm.setSize(275, 150);
jfrm.setLayout(new FlowLayout());
// to terminate on close
jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
// text label
JLabel jlab = new JLabel("Enter the divider and divident:");
// add text field for both numbers
JTextField ajtf = new JTextField(8);
JTextField bjtf = new JTextField(8);
// calc button
JButton button = new JButton("Calculate");
// labels
JLabel err = new JLabel();
JLabel alab = new JLabel();
JLabel blab = new JLabel();

JLabel anslab = new JLabel();
// add in order :
jfrm.add(err); // to display error bois
jfrm.add(jlab);
jfrm.add(ajtf);
jfrm.add(bjtf);
jfrm.add(button);
jfrm.add(alab);
jfrm.add(blab);
jfrm.add(anslab);
ActionListener l = new ActionListener() {
public void actionPerformed(ActionEvent evt) {
System.out.println("Action event from a text field");
}
};
ajtf.addActionListener(l);
bjtf.addActionListener(l);
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("\nA = " + a);
blab.setText("\nB = " + b);
anslab.setText("\nAns = " + ans);
}
catch(NumberFormatException e){
alab.setText("");
blab.setText("");
anslab.setText("");
err.setText("Enter Only Integers!");
}
catch(ArithmeticException e){
alab.setText("");
blab.setText("");
anslab.setText("");
err.setText("B should be NON zero!");
}
}
});
// display frame
jfrm.setVisible(true);
}
public static void main(String args[]){
// create frame on event dispatching thread
SwingUtilities.invokeLater(new Runnable(){
public void run(){
new SwingDemo();
}
}
);
}

```

