

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



LAB REPORT
on
Object Oriented Java Programming
(23CS3PCOOJ)

Submitted by

Adarsh Ganeshan(**1BM23CS014**)

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 **Adarsh Ganeshan (1BM23CS014)**, who is a bonafide student of **B.M.S. College of Engineering**. It is in partial fulfillment for the award of **Bachelor of Engineering in Computer Science and Engineering** of the Visvesvaraya Technological University, Belgaum. The Lab report has been approved as it satisfies the academic requirements in respect of an Object Oriented Java Programming (23CS3PCOOJ) work prescribed for the said degree.

Prof Geeta N. 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		Quadratic Equations	
2		Student Details	
3		Books Data	
4		Shapes	
5		Bank	
6		Student Marks	
7		Father-Son	
8		Threads	
9			
10		Division	

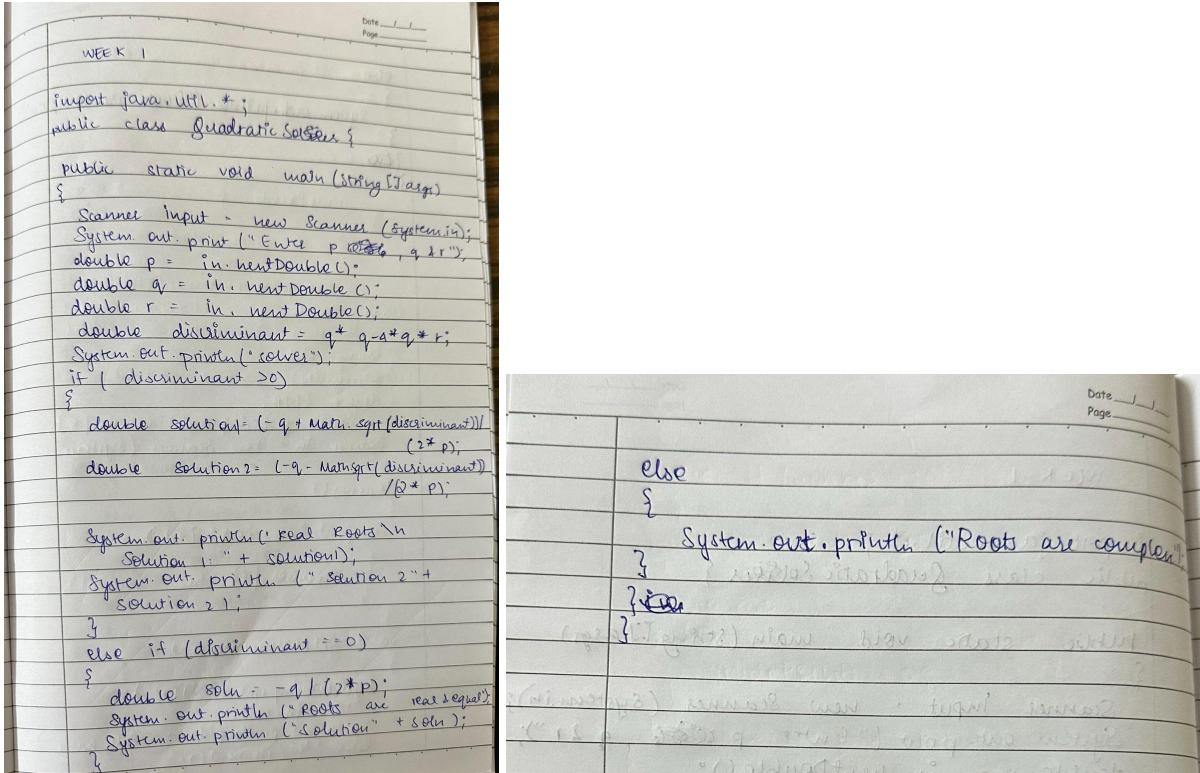
Github Link:

<https://github.com/AdarshG1125/OOJ/blob/main/WEEK1>

Program 1

Implement Quadratic Equation

Algorithm:



The image shows two pages of handwritten notes. The left page is titled 'WEEK 1' and contains the Java code for a quadratic solver. The right page continues the algorithm, showing the logic for complex roots and some handwritten notes.

```
WEEK 1

import java.util.*;
public class QuadraticSolver {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.print("Enter p: ");
        double p = input.nextDouble();
        double q = input.nextDouble();
        double r = input.nextDouble();
        double discriminant = q * q - 4 * p * r;
        System.out.println("solver");
        if (discriminant > 0) {
            double solution1 = (-q + Math.sqrt(discriminant)) / (2 * p);
            double solution2 = (-q - Math.sqrt(discriminant)) / (2 * p);
            System.out.println("Real Roots\nSolution 1: " + solution1);
            System.out.println("Solution 2: " + solution2);
        } else if (discriminant == 0) {
            double soln = -q / (2 * p);
            System.out.println("Roots are real & equal");
            System.out.println("Solution: " + soln);
        } else {
            System.out.println("Roots are complex");
        }
    }
}
```

else
System.out.println ("Roots are complex")
else if (discriminant == 0)
double soln = -q / (2 * p);
System.out.println ("Roots are real & equal");
System.out.println ("Solution: " + soln);
else
System.out.println ("Roots are complex");

Code:

```
import java.util.Scanner;

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

        System.out.print("Enter p: ");
        double p = input.nextDouble();
        System.out.print("Enter q: ");
        double q = input.nextDouble();
        System.out.print("Enter r: ");
        double r = input.nextDouble();

        double discriminant = q * q - 4 * p * r;

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

```
if (discriminant > 0) {  
    double solution1 = (-q + Math.sqrt(discriminant)) / (2 * p);  
    double solution2 = (-q - Math.sqrt(discriminant)) / (2 * p);  
  
    System.out.println("Real Roots");  
    System.out.println("Solution 1: " + solution1);  
    System.out.println("Solution 2: " + solution2);  
}  
else if (discriminant == 0) {  
    double solution = -q / (2 * p);  
    System.out.println("Roots are real and equal");  
    System.out.println("Solution: " + solution);  
}  
else {  
    System.out.println("Roots are complex");  
}  
  
input.close();  
}  
}
```

Github Link:

<https://github.com/AdarshG1125/OOJ/blob/main/WEEK2>

Program 2

Student Details

Algorithm:

WEEK 2

```
8 import java.util.*;  
class course  
{  
    int marks;  
    int creditHours;  
    int gradePoints;  
  
    void calculateGradePoints()  
    {  
        if (marks >= 90)  
        {  
            gradePoints = 10;  
        }  
        else if (marks >= 80)  
        {  
            gradePoints = 9;  
        }  
        else if (marks >= 70)  
        {  
            gradePoints = 8;  
        }  
        else if (marks >= 60)  
        {  
            gradePoints = 7;  
        }  
        else if (marks >= 50)  
        {  
            gradePoints = 6;  
        }  
    }  
}
```

```
else if (marks >= 40)  
{  
    gradePoints = 5;  
}  
else  
{  
    gradePoints = 0;  
}  
}  
  
class Learner  
{  
    String sId;  
    String fN;  
    double CGPA;  
    courses[] courses = new courses[8];  
    Scanner in = new Scanner (System.in);  
  
    learner()  
    {  
        for (int i=0; i<8; i++)  
        {  
            courses[i] = new courses();  
        }  
    }  
    void getLearnerDetails()  
    {  
        System.out.print ("SID?");  
        sId = in.next();  
        System.out.print ("Enter full name");  
        fN = in.next();  
    }  
}
```

```

    void getCourseMarks() {
        for (int i = 0; i < 8; i++) {
            System.out.print("Enter marks ");
            course[i].marks = in.nextInt();
            System.out.print("Enter credit ");
            course[i].creditHours = in.nextInt();
            course[i].calcGradePoints();
        }
    }

    void calculateCGPA() {
        double totalScore = 0;
        int totalCredits = 0;
        for (int i = 0; i < 8; i++) {
            totalScore += (course[i].gradePoints *
                           course[i].creditHours);
            totalCredits += course[i].creditHours;
        }
        CGPA = (totalCredits > 0) ? (totalScore / totalCredits) : 0;
    }

    void display() {
        System.out.println("StudentId " + studentId +
                           "\n" + fullName + "\n" + "CGPA");
    }
}

```

```

public class StudentRecords {
    public static void main (String [] args) {
        Learner [] learners = new Learner[3];
        for (int j = 0; j < 3; j++) {
            System.out.print("Enter details ");
            learners[j] = new Learner();
            learners[j].getLearnerDetails();
            learners[j].getCourseMarks();
            learners[j].cumulativeGPA();
        }
        for (int i = 0; i < 3; i++) {
            learners[i].display();
        }
    }
}

```

Code:

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

```

class Course {
    int marks;
    int creditHours;
    int gradePoints;

    void calculateGradePoints() {
        if (marks >= 90) {
            gradePoints = 10;
        } else if (marks >= 80) {
            gradePoints = 9;
        } else if (marks >= 70) {
            gradePoints = 8;
        } else if (marks >= 60) {
            gradePoints = 7;
        } else if (marks >= 50) {
            gradePoints = 6;
        } else if (marks >= 40) {
            gradePoints = 5;
        } else {
            gradePoints = 0;
        }
    }
}

```

```

    }

}

class Learner {
    String studentId;
    String fullName;
    double cumulativeGPA;
    Course[] courses = new Course[8];
    Scanner input = new Scanner(System.in);

    Learner() {
        for (int i = 0; i < 8; i++) {
            courses[i] = new Course();
        }
    }

    void getLearnerDetails() {
        System.out.print("Enter the student ID: ");
        studentId = input.next();
        System.out.print("Enter the full name: ");
        fullName = input.next();
    }

    void getCourseMarks() {
        for (int i = 0; i < 8; i++) {
            System.out.print("Enter marks for course " + (i + 1) + ": ");
            courses[i].marks = input.nextInt();
            System.out.print("Enter credit hours for course " + (i + 1) + ": ");
            courses[i].creditHours = input.nextInt();
            courses[i].calculateGradePoints();
        }
    }

    void calculateGPA() {
        double totalScore = 0;
        int totalCredits = 0;

        for (int i = 0; i < 8; i++) {
            totalScore += (courses[i].gradePoints * courses[i].creditHours);
            totalCredits += courses[i].creditHours;
        }
        cumulativeGPA = (totalCredits > 0) ? (totalScore / totalCredits) : 0;
    }

    void display() {
        System.out.println("Student ID: " + studentId);
        System.out.println("Full Name: " + fullName);
    }
}

```

```
        System.out.println("Cumulative GPA: " + cumulativeGPA);
    }
}

public class StudentRecords {
    public static void main(String[] args) {
        Learner[] learners = new Learner[3];

        for (int j = 0; j < 3; j++) {
            System.out.println("Enter the details for learner " + (j + 1) + ":");
            learners[j] = new Learner();
            learners[j].getLearnerDetails();
            learners[j].getCourseMarks();
            learners[j].calculateGPA();
        }

        for (int i = 0; i < 3; i++) {
            learners[i].display();
        }
    }
}
```

Github Link:

<https://github.com/AdarshG1125/OOJ/blob/main/WEEK3>

Program 3

Books Data

Algorithm:

The image shows handwritten Java code on two pages. The first page contains the definition of the `Book` class, which has attributes `name`, `author`, `price`, and `numPages`. It also includes a constructor and a `toString` method. The second page contains the `main` method of the `Book` class, which reads input from the user and prints the details of each book.

```
import java.util.*;  
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()  
    {  
        String bookDetails =  
            "Book Name:" + this.name +  
            "Author Name:" + this.author +  
            "Price :" + this.price +  
            "Number of pages :" + this.numPages;  
        return bookDetails;  
    }  
}  
  
public class Main  
{  
    public static void main (String [] args)  
    {  
        Scanner in = new Scanner (System.in);  
        System.out.println ("Enter the no. of books");  
        int n = in.nextInt();  
        Book [] books = new Book [n];  
  
        for (int i=0; i<n; i++)  
        {  
            System.out.println ("Name of book" +  
                (i+1) + ":");  
            String name = in.nextLine();  
            System.out.println ("Author" + (i+1) + ":");  
            String author = in.nextLine();  
            System.out.println ("Price ?" + (i+1));  
            int price = in.nextInt();  
            System.out.println ("Pages ?" + (i+1));  
            int numPages = in.nextInt();  
            books[i] = new Book (name,  
                               author,  
                               price,  
                               numPages);  
        }  
    }  
}
```

The image shows the execution of the program. The user is prompted to enter the number of books, which is 1. Then, the user enters the details for one book: Name (Book), Author (me), Price (1), and Pages (1). The program then prints the `toString` representation of the book, which includes all four attributes: Name, Author, Price, and Pages.

```
System.out.println ("Enter the no. of books");  
int n = in.nextInt();  
  
for (int i=0; i<n; i++)  
{  
    System.out.println ("Name of book" + (i+1) + ":");  
    String name = in.nextLine();  
    System.out.println ("Author" + (i+1) + ":");  
    String author = in.nextLine();  
    System.out.println ("Price ?" + (i+1));  
    int price = in.nextInt();  
    System.out.println ("Pages ?" + (i+1));  
    int numPages = in.nextInt();  
    books[i] = new Book (name,  
                        author,  
                        price,  
                        numPages);  
}  
  
for (int i=0; i<n; i++)  
{  
    System.out.println (books[i]);  
}
```

OUTPUT:
Enter the number of books: 1
Name of book: Book
Author: me
Price: 1
Pages: 1

BOOK Details:
BOOK Name: Book
AUTHOR name: me
Price : 1
No. of pages : 1

Code:

```
import java.util.Scanner;

class Publication {
    String title;
    String writer;
    int cost;
    int pageCount;

    Publication(String title, String writer, int cost, int pageCount) {
        this.title = title;
        this.writer = writer;
        this.cost = cost;
        this.pageCount = pageCount;
    }

    @Override
    public String toString() {
        String publicationDetails = "Title: " + this.title + "\n" +
            "Author: " + this.writer + "\n" +
            "Cost: " + this.cost + "\n" +
            "Pages: " + this.pageCount + "\n";
        return publicationDetails;
    }
}

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

        System.out.print("Enter the Number of Publications: ");
        int count = input.nextInt();

        Publication[] publications = new Publication[count];

        for (int i = 0; i < count; i++) {
            System.out.print("Enter title of publication " + (i + 1) + ": ");
            String title = input.next();
            System.out.print("Enter author of publication " + (i + 1) + ": ");
            String author = input.next();
            System.out.print("Enter cost of publication " + (i + 1) + ": ");
            int cost = input.nextInt();
            System.out.print("Enter number of pages in publication " + (i + 1) + ": ");
            int pageCount = input.nextInt();

            publications[i] = new Publication(title, author, cost, pageCount);
        }
    }
}
```

```
System.out.println("\nPublication Details:");
for (Publication publication : publications) {
    System.out.println(publication);
}
input.close();
}
```

Github Link:

<https://github.com/AdarshG1125/OOJ/blob/main/WEEK4>

Program 4

Shapes

Algorithm:

```
import java.util.*;
abstract class Shape
{
    abstract void printArea();
}

class Rectangle extends Shape
{
    private int w, h;
    Rectangle (int w, int h)
    {
        this.w = w;
        this.h = h;
    }
    void printArea()
    {
        System.out.println("Area of rectangle:" + (w * h));
    }
}
```

```
class Triangle extends Shape
{
    private int b, h;
    Triangle (int b, int h)
    {
        this.b = b;
        this.h = h;
    }
    void printArea()
    {
        System.out.println("Area of triangle:" + (0.5 * b * h));
    }
}

class Circle extends Shape
{
    private int r;
    Circle (int r)
    {
        this.r = r;
    }
    void printArea()
    {
        System.out.println("Area of circle:" + (3.14 * r * r));
    }
}

public class Main
{
    public static void main (String [] args)
    {
        Scanner sc = new Scanner (System.in);
    }
}
```

```
System.out.println("Enter width & height of Rectangle");
Shape rect = new Rectangle (sc.nextInt(),
                           sc.nextInt());

System.out.println("Enter base & height of Triangle");
Shape tri = new Triangle (sc.nextInt(),
                         sc.nextInt());
System.out.println("Enter radius of circle");
Shape circ = new Circle (sc.nextInt());

rect.printArea();
tri.printArea();
circ.printArea();

OUTPUT
Enter width & height of Rectangle: 60
Enter base & height of Triangle: 20
Enter radius of circle: 50
Area of Rectangle: 3000
Area of Triangle: 500.0
Area of Circle: 7850.0
```

Code:

```
import java.util.Scanner;

abstract class GeometricShape {
    int dimension1;
    int dimension2;

    public GeometricShape() {
        this.dimension1 = 0;
        this.dimension2 = 0;
    }

    public GeometricShape(int dimension1, int dimension2) {
        this.dimension1 = dimension1;
        this.dimension2 = dimension2;
    }

    public abstract void calculateArea();
}

class RectangleShape extends GeometricShape {
    public RectangleShape(int length, int width) {
        dimension1 = length;
        dimension2 = width;
    }

    public void calculateArea() {
        int area = dimension1 * dimension2;
        System.out.println("Area of Rectangle: " + area);
    }
}

class TriangleShape extends GeometricShape {
    public TriangleShape(int base, int height) {
        dimension1 = base;
        dimension2 = height;
    }

    public void calculateArea() {
        double area = 0.5 * dimension1 * dimension2;
        System.out.println("Area of Triangle: " + area);
    }
}

class CircleShape extends GeometricShape {
    public CircleShape(int radius) {
        dimension1 = radius;
    }
}
```

```

        dimension2 = 0;
    }

    public void calculateArea() {
        double area = Math.PI * dimension1 * dimension1;
        System.out.println("Area of Circle: " + area);
    }
}

public class ShapeCalculator {

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

        System.out.println("Enter length and width for Rectangle:");
        int length = input.nextInt();
        int width = input.nextInt();
        GeometricShape rectangle = new RectangleShape(length, width);
        rectangle.calculateArea();

        System.out.println("Enter base and height for Triangle:");
        int base = input.nextInt();
        int height = input.nextInt();
        GeometricShape triangle = new TriangleShape(base, height);
        triangle.calculateArea();

        System.out.println("Enter radius for Circle:");
        int radius = input.nextInt();
        GeometricShape circle = new CircleShape(radius);
        circle.calculateArea();

        input.close();
    }
}

```

Github Link:

<https://github.com/AdarshG1125/OOJ/blob/main/WEEK5>

Program 5

Bank

Algorithm:

```
import java.util.*;  
class Account {  
    protected String customerName;  
    protected int accountNumber;  
    protected double balance;  
    public Account( String customerName,  
        int accountNumber, double balance )  
    {  
        this.customerName = customerName;  
        this.accountNumber = accountNumber;  
        this.balance = balance;  
    }  
    public void deposit( double amount )  
    {  
        if ( amount > 0 )  
        {  
            balance += amount;  
            System.out.println( " Deposited : " + amount );  
        }  
        else  
        {  
            System.out.println( " Invalid ! " );  
        }  
    }  
}
```

```
public void displayBalance()  
{  
    System.out.println( " Balance : " + balance );  
}  
class SavAcct extends Account  
{  
    private double interestRate;  
    public SavAcct( String customerName,  
        int accountNumber, double balance, double interestRate )  
    {  
        super( customerName, accountNumber, balance );  
        this.interestRate = interestRate;  
    }  
    public void compDepInterest()  
    {  
        double interest = balance * ( interestRate / 100 );  
        balance -= amount;  
        System.out.println( " Withdrawn : " + amount );  
        System.out.println( " Interest added : " + interest );  
    }  
    public void withdraw( double amount )  
    {  
        if ( amount <= balance )  
        {  
            balance -=
```

Code:

```
import java.util.Scanner;  
  
class BankAccount {  
    protected String accountHolderName;  
    protected int accountId;  
    protected double accountBalance;  
  
    public BankAccount( String accountHolderName, int accountId, double accountBalance ) {  
        this.accountHolderName = accountHolderName;  
        this.accountId = accountId;  
        this.accountBalance = accountBalance;  
    }  
  
    public void depositAmount( double amount ) {  
        if ( amount > 0 ) {  
            accountBalance += amount;
```

```

        System.out.println("Deposited: " + amount);
    } else {
        System.out.println("Invalid deposit amount");
    }
}

public void showBalance() {
    System.out.println("Balance: " + accountBalance);
}
}

class SavingsAccount extends BankAccount {
    private double rateOfInterest;

    public SavingsAccount(String accountHolderName, int accountId, double accountBalance, double rateOfInterest) {
        super(accountHolderName, accountId, accountBalance);
        this.rateOfInterest = rateOfInterest;
    }

    public void calculateAndAddInterest() {
        double interest = accountBalance * (rateOfInterest / 100);
        accountBalance += interest;
        System.out.println("Interest added: " + interest);
    }

    public void withdrawAmount(double amount) {
        if (amount <= accountBalance) {
            accountBalance -= amount;
            System.out.println("Withdrawn: " + amount);
        } else {
            System.out.println("Insufficient balance for withdrawal");
        }
    }
}

class CurrentAccount extends BankAccount {
    private double minimumRequiredBalance;
    private double feeCharge;

    public CurrentAccount(String accountHolderName, int accountId, double accountBalance, double minimumRequiredBalance, double feeCharge) {
        super(accountHolderName, accountId, accountBalance);
        this.minimumRequiredBalance = minimumRequiredBalance;
        this.feeCharge = feeCharge;
    }
}

```

```

public void withdrawAmount(double amount) {
    if (amount <= accountBalance) {
        accountBalance -= amount;
        System.out.println("Withdrawn: " + amount);

        if (accountBalance < minimumRequiredBalance) {
            accountBalance -= feeCharge;
            System.out.println("Fee charge imposed: " + feeCharge);
        }
    } else {
        System.out.println("Insufficient balance for withdrawal");
    }
}

public class BankApplication {
    public static void main(String[] args) {
        Scanner inputScanner = new Scanner(System.in);
        SavingsAccount savingsAccount = new SavingsAccount("Alice", 12345, 1000, 5);
        CurrentAccount currentAccount = new CurrentAccount("Bob", 67890, 2000, 500, 50);

        System.out.println("Choose Account Type:\n1. Savings Account\n2. Current Account");
        int userChoice = inputScanner.nextInt();

        switch (userChoice) {
            case 1:
                System.out.println("Savings Account Selected");
                savingsAccount.depositAmount(500);
                savingsAccount.calculateAndAddInterest();
                savingsAccount.withdrawAmount(300);
                savingsAccount.showBalance();
                break;

            case 2:
                System.out.println("Current Account Selected");
                currentAccount.depositAmount(500);
                currentAccount.withdrawAmount(1800);
                currentAccount.showBalance();
                break;

            default:
                System.out.println("Invalid choice");
        }

        inputScanner.close();
    }
}

```

Github Link:

<https://github.com/AdarshG1125/OOJ/blob/main/WEEK6>

Program 6

Student Marks

Algorithm:

```
Week 6
package CIE;
// saving in file CIE/Internal
public class Internal extends Student {
    public int[] internalMarks;
    public Internal (String usn, String name, int sem, int[] internalMarks) {
        super (usn, name, sem);
        this.internalMarks = internalMarks;
    }
}

package SEE;
// saving in file SEE/ External.java
import CIE.Student;

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

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

public class FinalMarks {
    public static void main (String args) {
        Scanner in = new Scanner (System.in);
        System.out.println ("Enter number of students");
        int n = in.nextInt();
        Internal[] internalStudents = new Internal [n];
        External[] externalStudents = new External [n];
        for (int i=0; i<n; i++) {
            System.out.print ("Enter details");
            System.out.print ("Name : ");
            String name = in.nextLine();
            System.out.print ("Semesters : ");
            int sem = in.nextInt();
            int[] internalMarks = new int [5];
            System.out.print ("Enter internal marks for 5 courses");
            for (int j=0; j<5; j++) {
                internalMarks [j] = in.nextInt();
            }
        }
    }
}
```

```
for (int j=0; j<5; j++) {
    Internal Marks [j] = in.nextInt();
}
int[] seeMarks = new int [5];
System.out.print ("Enter SEE marks for 5 courses");
for (int j=0; j<5; j++) {
    seeMarks [j] = in.nextInt();
}

// Objects for Internals & External
// created here
internalStudents [i] = new Internal (usn, name, sem, internalMarks);
externalStudents [i] = new External (usn, name, sem, seeMarks);
}

System.out.println ("Final marks of student");
for (int i=0; i<n; i++) {
    System.out.print ("Student " + (i+1) + " usn");
    + internalStudents [i].usn);
    for (int j=0; j<5; j++) {
        int finalMark =
            internalStudents [i].internalMarks [j]
            + (externalStudents [i].seeMarks [j]/2));
        System.out.print ("Course " + (j+1) +
            " Final Marks : " + finalMark);
    }
}
```

```
} Scanner in.close
}

package CIE;
public class Student {
    // stored in CIE / Student.java.
    public String usn;
    public String name;
    public int semester;
    public Student (String usn, String name, int sem) {
        this.usn = usn;
        this.name = name;
        this.semester = sem;
    }
}
```

OUTPUT

```

Enter number of students : 1
Enter details for student 1:
USN: 1BM23CS014
Name: Adarsh
Semester: 3
Enter internal marks for 5 courses:
50
40
32
36
40
Enter SEE marks for 5 courses:
80
80
80
80
80
See Final Marks of students
Student 1 USN: 1BM23CS014
Course 1 Final Mark: 90
Course 2 Final Mark: 80
Course 3 Final Mark: 72
Course 4 Final Mark: 376
Course 5 Final Mark: 80

```

Code:

CIE/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.print("Enter USN: ");
        usn = s.nextLine();
        System.out.print("Enter Name: ");
        name = s.nextLine();
        System.out.print("Enter Semester: ");
        sem = s.nextInt();
    }
    public void displayStudentDetails() {
        System.out.println("USN: " + usn);
        System.out.println("Name: " + name);
        System.out.println("Semester: " + sem);
    }
}

```

CIE/Internals.java

```

package CIE;
import java.util.Scanner;
public class Internals extends Student {

```

```

protected int[] internalMarks = new int[5];
public void inputCIEmarks() {
Scanner s = new Scanner(System.in);
System.out.println("Enter Internal Marks for 5 subjects:");
for (int i = 0; i < 5; i++) {
System.out.print("Subject " + (i + 1) + ": ");
internalMarks[i] = s.nextInt();
}
}
}
}

```

SEE/Externals.java

```

package SEE;
import CIE.Internals;
import java.util.Scanner;
public class Externals extends Internals {
private int[] seeMarks = new int[5];
private int[] finalMarks = new int[5];
public void inputSEEmarks() {
Scanner s = new Scanner(System.in);
System.out.println("Enter SEE Marks for 5 subjects:");
for (int i = 0; i < 5; i++) {
System.out.print("Subject " + (i + 1) + ": ");
seeMarks[i] = s.nextInt();
}
}
}
public void calculateFinalMarks() {
for (int i = 0; i < 5; i++) {
finalMarks[i] = internalMarks[i] + seeMarks[i];
}
}
}
public void displayFinalMarks() {
displayStudentDetails();
System.out.println("Final Marks for 5 subjects:");
for (int i = 0; i < 5; i++) {
System.out.println("Subject " + (i + 1) + ": " + finalMarks[i]);
}
}
}
}

```

Main.java

```

import SEE.Externals;
import java.util.Scanner;
class Main {

```

```
public static void main(String[] args) {  
    Scanner s = new Scanner(System.in);  
    System.out.print("Enter number of students: ");  
    int n = s.nextInt();  
    Externals[] students = new Externals[n];  
    for (int i = 0; i < n; i++) {  
        System.out.println("\nEnter details for student " + (i + 1) + ":");  
        students[i] = new Externals();  
        students[i].inputStudentDetails();  
        students[i].inputCIEmarks();  
        students[i].inputSEEmarks();  
        students[i].calculateFinalMarks();  
    }  
    System.out.println("\nFinal Marks of Students:");  
    for (int i = 0; i < n; i++) {  
        System.out.println("\nStudent " + (i + 1) + ":");  
        students[i].displayFinalMarks();  
    }  
}
```

Github Link:

<https://github.com/AdarshG1125/OOJ/blob/main/WEEK7>

Program 7

Father-Son

Algorithm:

Q WAP that demonstrates handling of exception in inheritance prog.

```
class WrongAgeException extends Exception
{
    public WrongAgeException (String msg)
    {
        super (msg);
    }

    class Father son extends Father
    {
        int age;
        public Father (int age) throws WrongAgeException
        {
            super (age);
            if (age < 0)
                throw new WrongAgeException ("Son's age can't be negative");
            if (sAge > fAge)
                throw new WrongAgeException ("Son's age can't be greater than father's age");
        }
    }
}
```

Date / /
Page / /

```
this.sAge = sAge;
System.out.println ("Son's age : " + sAge);
}

public class Father
{
    int age;
    public Father (int age) throws WrongAgeException
    {
        if (age < 0)
            throw new WrongAgeException ("Father's age can't be negative");
        this.age = age;
        System.out.println ("Father's age : " + age);
    }
}
```

```

public class Main {
    public static void main (String args) {
        Scanner sc = new Scanner (System.in);
        try {
            System.out.println ("Father's age");
            int fAge = sc.nextInt();
            System.out.println ("Son's age");
            int sAge = sc.nextInt();
            Son son = new Son (fAge, sAge);
        } catch (WrongAgeException e) {
            System.out.println ("Exception: " + e.getMessage());
        } catch (Exception e) {
            System.out.println ("Invalid input.");
        }
    }
}

OUTPUT
Enter father's age: 20
Enter son's age: 44
Father's age: 20
Exception: Son's age can't be
greater than father's age.

```

Code:

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

```

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

class Parent {
    int parentAge;

    public Parent(int parentAge) throws InvalidAge {
        if (parentAge < 0) {
            throw new InvalidAge("Age Cannot be Negative");
        }
        this.parentAge = parentAge;
        System.out.println("Parent's Age: " + this.parentAge);
    }
}

class Child extends Parent {
    int childAge;
}

```

```

public Child(int parentAge, int childAge) throws InvalidAge {
    super(parentAge);
    if (childAge < 0) {
        throw new InvalidAge("Child's Age Cannot be Negative");
    }
    if (childAge >= parentAge) {
        throw new InvalidAge("Child's Age Cannot be Greater than or Equal to Parent's Age");
    }
    this.childAge = childAge;
    System.out.println("Child's Age: " + this.childAge);
}
}

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

        System.out.print("Enter Parent's Age: ");
        int parentAge = inputScanner.nextInt();

        System.out.print("Enter Child's Age: ");
        int childAge = inputScanner.nextInt();

        try {
            Child child = new Child(parentAge, childAge);
        } catch (InvalidAge e) {
            System.out.println("Exception: " + e.getMessage());
        }
        inputScanner.close();
    }
}

```

Github Link:

<https://github.com/AdarshG1125/OOJ/blob/main/WEEK8>

Program 8

Threads

Algorithm:

```
public class BMSCollege {
    public static void main (String [] args) {
        Thread t1 = new Thread () -> {
            while (true) {
                System.out.println ("BMS college of Eng");
                try { Thread.sleep (1000); }
                catch (InterruptedException e) {}
            }
        };
        Thread t2 = new Thread () -> {
            while (true) {
                System.out.println ("CSE");
                try { Thread.sleep (2000); }
                catch (InterruptedException e) {}
            }
        };
        t1.start ();
        t2.start ();
        try {
            Thread.sleep (2000);
        } catch (InterruptedException e) {}
        System.exit (0);
    }
}
```

OUTPUT

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

Code:

```
class CollegeThread extends Thread {  
    public void run() {  
        try {  
            for (int i = 0; i < 5; i++) {  
                System.out.println("BMS College of Engineering");  
                Thread.sleep(10000);  
            }  
        } catch (InterruptedException e) {  
            System.out.println("CollegeThread interrupted.");  
        }  
    }  
}  
  
class CSEThread extends Thread {  
    public void run() {  
        try {  
            for (int i = 0; i < 25; i++) {  
                System.out.println("CSE");  
                Thread.sleep(2000);  
            }  
        } catch (InterruptedException e) {  
            System.out.println("CSEThread interrupted.");  
        }  
    }  
}  
  
public class ThreadExample {  
    public static void main(String[] args) {  
        CollegeThread collegeThread = new CollegeThread();  
        CSEThread cseThread = new CSEThread();  
  
        collegeThread.start();  
        cseThread.start();  
    }  
}
```

Github Link:

<https://github.com/AdarshG1125/OOJ/blob/main/WEEK9>

Program 9

Implement Quadratic Equation

Algorithm:

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");  
        }  
  
        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 A.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[]) {
    new Deadlock();
}
}
```

Github Link:

<https://github.com/AdarshG1125/OOJ/blob/main/WEE10>

Program 10

Division

Algorithm:

Code:

```
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 divider and divident:");
        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(err);
        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 (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();
        }
    });
}
}

```