Course Name:	**Object Oriented Programming Methodology**	**Semester:**	**III**
Date of Performance:	___ / ___ / _____	**Batch No:**	**B2**
Faculty Name:	**Prof. Kiran Thale**	**Roll No:**	**16010122151**
Faculty Sign & Date:		**Grade/Marks:**	___/25

Experiment No: 6

Title: Case Study (Class Diagram)

Aim and Objective of the Experiment:

Draw class Diagram for the chosen Case Study. Clearly show

- Attributes
- Multiplicities between classes
- Aggregations/compositions/Association between classes
- Generalization between classes in the class diagram.

And show the implementation of aggregation, association, composition and generalization between the classes.

COs to be achieved:

CO1: Understand the features of object oriented programming compared with procedural approach with C++ and Java.
CO2: Explore arrays, vectors, classes and objects in C++ and Java.
CO3: Implement scenarios using object oriented concepts (Drawing class diagram, relationship between classes, sequence diagram)
CO4: Explore the interface, exceptions, multithreading, packages

Tools used:

JDK, VScode / Eclipse

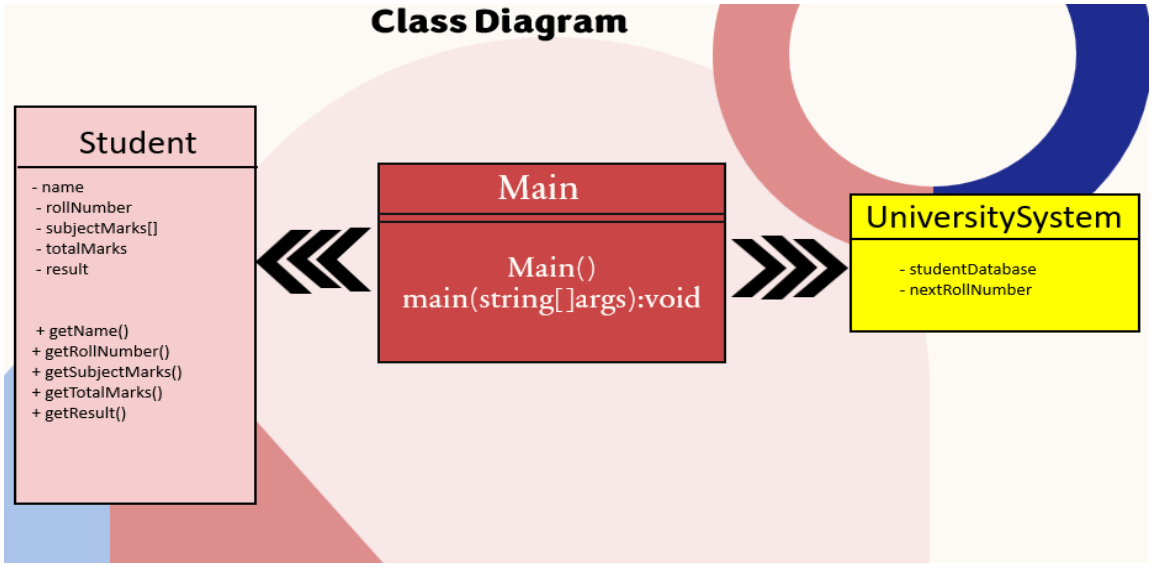
Theory:

Pre Lab/ Prior Concepts:

Define Class, Methods and Object.
Understanding of Aggregation, Association, Composition and Generalization between classes

List Of Classes:
Identify Attributes for each class:
Identify List of Methods in each classes:

Class Diagram:



Algorithm:

Algorithm for University System Program:

1. Initialize a map called `studentDatabase` to store student records and an integer `nextRollNumber` to assign roll numbers to new students. Set `nextRollNumber` to 1.
2. Create the `UniversitySystem` class with the `main` method:
   - Initialize a variable `choice` to 0 to represent the user's menu choice.
   - Create a `Scanner` object named `scanner` for user input.
   - Enter a loop that continues until the user chooses to exit (choice equals 3).
3. Inside the loop:
   - Display the main menu with three options: Administrator, Student, and Exit.
   - Prompt the user to enter a choice.
   - Use a try-catch block to handle exceptions, ensuring that the user's input is a valid integer.
4. Based on the user's choice:
   - If the choice is 1 (Administrator), call the `administerSystem` method with the `scanner` object.
   - If the choice is 2 (Student), call the `studentSystem` method with the `scanner` object.
   - If the choice is 3 (Exit), display an exit message and exit the program.

7. Define a `Student` class to encapsulate student information, including name, roll number, subject marks, total marks, and result.
8. Create getter methods within the `Student` class to access its attributes.
9. The program continues to run until the user selects the "Exit" option in the main menu.

This algorithm outlines the main functionality and flow of the University System program. It allows administrators to add student records and students to view their own information, all while ensuring data integrity and error handling.

Code:

```java
import java.io.*;
import java.util.HashMap;
import java.util.Map;
import java.util.Scanner;

public class UniversitySystem {
    private static Map<Integer, Student> studentDatabase = new HashMap<>();
    private static int nextRollNumber = 1;

    public static void main(String[] args) {
        int choice = 0;
        Scanner scanner = new Scanner(System.in);

        while (choice != 3) {
            System.out.println("Welcome to the University System");
            System.out.println("1. Administrator");
            System.out.println("2. Student");
            System.out.println("3. Exit");
            System.out.print("Enter your choice: ");

            try {
                choice = scanner.nextInt();
                scanner.nextLine();

                switch (choice) {
                    case 1:
                        administerSystem(scanner);
                        break;
                    case 2:
                        studentSystem(scanner);
                        break;
                    case 3:
                        System.out.println("Exiting the University System.");
                        break;
                    default:
                        System.out.println("Invalid choice. Please select 1, 2, or 3.");
                }
            } catch (Exception e) {
                System.out.println("Invalid input. Please enter a valid choice.");
                scanner.nextLine();
            }
        }
        scanner.close();
    }

    private static void administerSystem(Scanner scanner) {
```

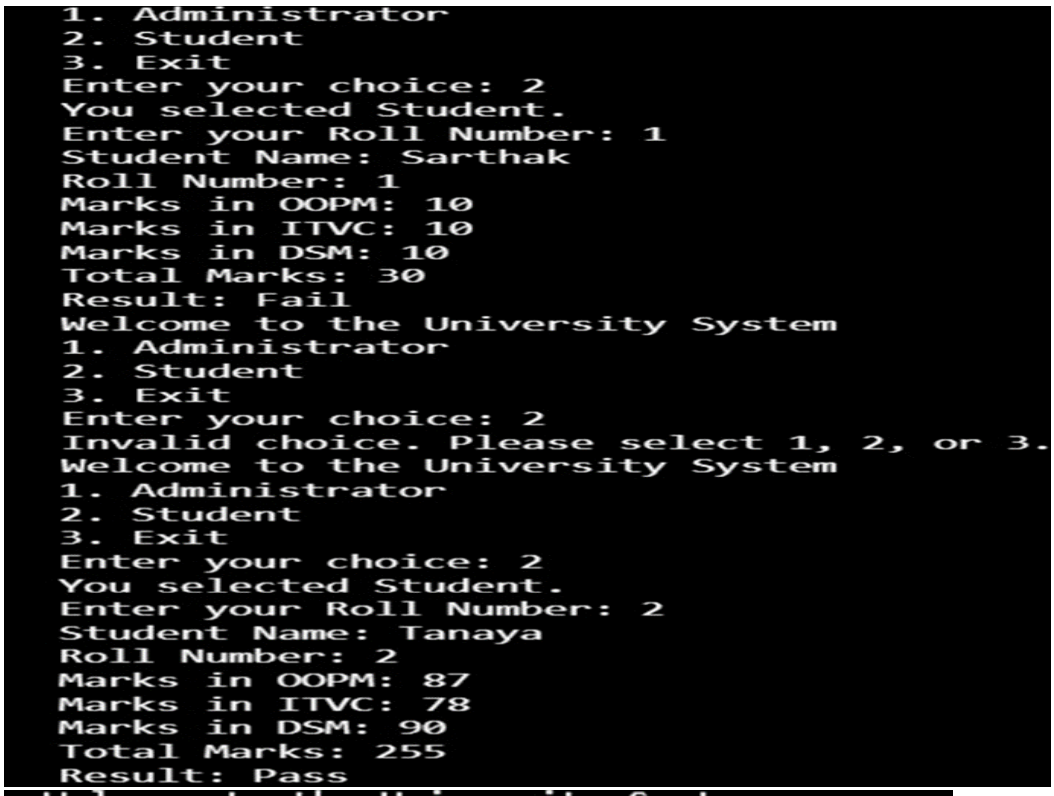
```
        System.out.print("Enter Student Name: ");
        String studentName = scanner.nextLine();
        int rollNumber = nextRollNumber++;


        int[] subjectMarks = new int[3];
        String[] subjectNames = {"OOPM", "ITVC", "DSM"};

        for (int i = 0; i < 3; i++) {
            System.out.print("Enter Marks in " + subjectNames[i] + ": ");
            subjectMarks[i] = scanner.nextInt();
        }

        int totalMarks = 0;
        for (int marks : subjectMarks) {
            totalMarks += marks;
        }

        String result = (totalMarks >= 50) ? "Pass" : "Fail";

        Student student = new Student(studentName, rollNumber, subjectMarks, totalMarks, result);
        studentDatabase.put(rollNumber, student);


            System.out.println("Student added to the database with Roll Number: " + rollNumber);
    }

    private static void studentSystem(Scanner scanner) {
        System.out.println("You selected Student.");
        System.out.print("Enter your Roll Number: ");
        int rollNumber = scanner.nextInt();

        if (studentDatabase.containsKey(rollNumber)) {
            Student student = studentDatabase.get(rollNumber);
            System.out.println("Student Name: " + student.getName());
            System.out.println("Roll Number: " + student.getRollNumber());

            int[] subjectMarks = student.getSubjectMarks();
            String[] subjectNames = {"OOPM", "ITVC", "DSM"};
            for (int i = 0; i < 3; i++) {
                System.out.println("Marks in " + subjectNames[i] + ": " + subjectMarks[i]);
            }

            System.out.println("Total Marks: " + student.getTotalMarks());
            System.out.println("Result: " + student.getResult());
        } else {
            System.out.println("Student with Roll Number " + rollNumber + " not found.");
```

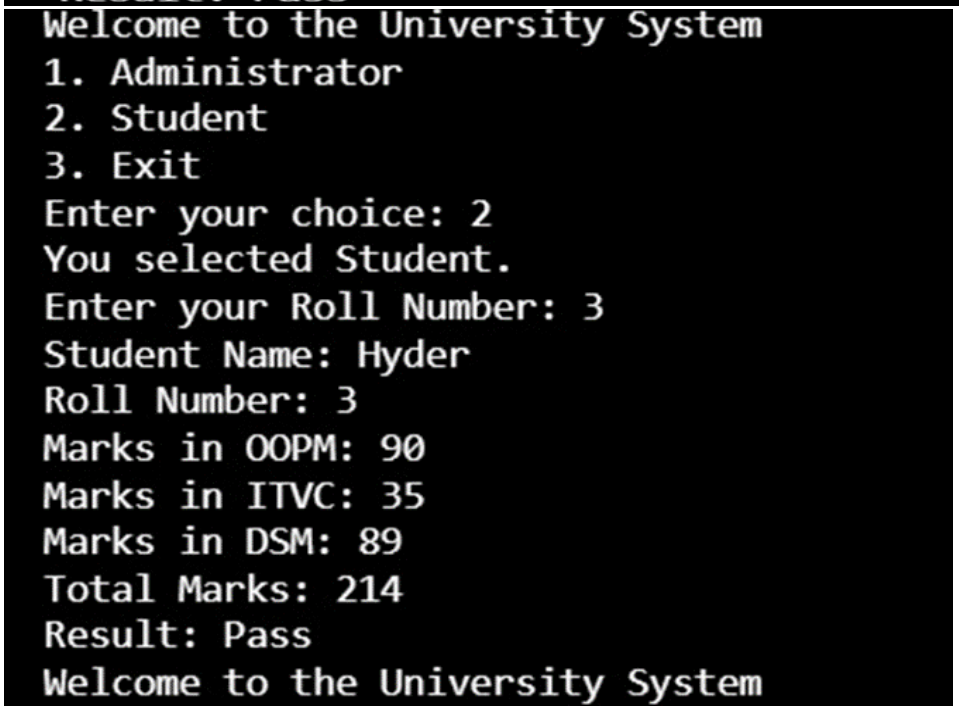
```
97              }
98          }
99      }
100
101     class Student {
102         private String name;
103         private int rollNumber;
104         private int[] subjectMarks;
105         private int totalMarks;
106         private String result;
107
108         public Student(String name, int rollNumber, int[] subjectMarks, int totalMarks, String result) {
109             this.name = name;
110             this.rollNumber = rollNumber;
111             this.subjectMarks = subjectMarks;
112             this.totalMarks = totalMarks;
113             this.result = result;
114         }
115
116         public String getName() {
117             return name;
118         }
```

```
119
120         public int getRollNumber() {
121             return rollNumber;
122         }
123
124         public int[] getSubjectMarks() {
125             return subjectMarks;
126         }
127
128         public int getTotalMarks() {
129             return totalMarks;
130         }
131
132         public String getResult() {
133             return result;
134         }
135     }
```

Output:

```
Welcome to the University System
1. Administrator
2. Student
3. Exit
Enter your choice: 1
You selected Administrator.
Enter Student Name: Sarthak
Enter Marks in OOPM: 35
Enter Marks in ITVC: 34
Enter Marks in DSM: 20
Student added to the database with Roll Number: 1
Welcome to the University System
1. Administrator
2. Student
3. Exit
Enter your choice: 1
You selected Administrator.
Enter Student Name: Tanaya
Enter Marks in OOPM: 87
Enter Marks in ITVC: 78
Enter Marks in DSM: 90
Student added to the database with Roll Number: 2
Welcome to the University System
1. Administrator
2. Student
3. Exit
Enter your choice: 1
You selected Administrator.
Enter Student Name: Hyder
Enter Marks in OOPM: 90
Enter Marks in ITVC: 35
Enter Marks in DSM: 89
Student added to the database with Roll Number: 3
```


1. Administrator
2. Student
3. Exit
Enter your choice: 2
You selected Student.
Enter your Roll Number: 1
Student Name: Sarthak
Roll Number: 1
Marks in OOPM: 10
Marks in ITVC: 10
Marks in DSM: 10
Total Marks: 30
Result: Fail
Welcome to the University System
1. Administrator
2. Student
3. Exit
Enter your choice: 2
Invalid choice. Please select 1, 2, or 3.
Welcome to the University System
1. Administrator
2. Student
3. Exit
Enter your choice: 2
You selected Student.
Enter your Roll Number: 2
Student Name: Tanaya
Roll Number: 2
Marks in OOPM: 87
Marks in ITVC: 78
Marks in DSM: 90
Total Marks: 255
Result: Pass


Welcome to the University System
1. Administrator
2. Student
3. Exit
Enter your choice: 2
You selected Student.
Enter your Roll Number: 3
Student Name: Hyder
Roll Number: 3
Marks in OOPM: 90
Marks in ITVC: 35
Marks in DSM: 89
Total Marks: 214
Result: Pass
Welcome to the University System

Post Lab Subjective/Objective type Questions:

1) **1. Consider the following class:**

```
public class TypeOfVariable{
public static int a;
int b,c;
public void printValue(){
     int x = 10;
}
public static void main(String args[]){
     TypeOfVariable object=new TypeOfVariable();
     object.printValue();
}
}
```

a) What are the class/static variables?

int a;

b) What are the instance variables?

int b,c;

c) What are local variables?

```
int x;
```

2) **What is the output from the following code:**

```
public class Test
{
static int x = 11;
    private int y = 33;
    public void method1(int x)
    {
        Test t = new Test();
        this.x = 22;
        y = 44;
```

```
        System.out.println("Test.x: " + Test.x);
        System.out.println("t.x: " + t.x);
        System.out.println("t.y: " + t.y);
        System.out.println("y: " + y);
    }

    public static void main(String args[])
    {
        Test t = new Test();
        t.method1(5);
    }
}
```

```
Test.x:  22
t.x:  22
t.y:  33
y:  44
```

Conclusion:
We've learned how to use Object-Oriented Programming to solve real-life problems.

Signature of faculty in-charge with Date: