Vidyavardhini's College Of Engineering & Technology, Vasai

ASSIGNMENT NO. 2

NAME: CHETAN BHUYAL **ROLL NO.** : 16
DATE : **SUBJECT :** FULL STACK JAVA PROGRAMING

QUESTION

Q1. In a university's educational system, there are different levels of individuals associated with academics.

1. Person Class – Base Class (Top Level)

This class represents any human being involved in the system. It contains general personal information such as:

- **name: The person's full name.**
- **age: The person's age.**

2. Student Class – Inherits from Person

This class represents a student, who is also a person. So, it inherits all properties of Person, and adds more specific details:

- **rollNumber: Unique ID assigned to the student.**
- **course: The course or degree the student is enrolled in.**

3. GraduateStudent Class – Inherits from Student

This class represents a graduate student, who is a specialized kind of student. In addition to all details from Student and Person, this class adds:

- **specialization: The field in which the student is specializing.**
- **researchTopic: The topic or title of their thesis or research.**

Key Concepts Demonstrated:

- **A child class inherits from a class which is already a child of another class.**
- **Constructor Chaining: super() is used to initialize parent class data.**

- **Method Reuse: Methods are reused across the hierarchy (displayPersonInfo, displayStudentInfo, etc.). Apply**

CODE

```
// Q1: University Educational System Example
public class Main {

    // --- Base Class: Person ---
    static class Person {
        String name;
        int age;

        // Constructor
        Person(String name, int age) {
            this.name = name;
            this.age = age;
        }

        void displayPersonInfo() {
            System.out.println("Name: " + name);
            System.out.println("Age: " + age);
        }
    }

    // --- Derived Class: Student ---
    static class Student extends Person {
        String rollNumber;
        String course;

        // Constructor with super() to call parent constructor
```

```
    Student(String name, int age, String rollNumber, String
course) {
        super(name, age);
        this.rollNumber = rollNumber;
        this.course = course;
    }

    void displayStudentInfo() {
        displayPersonInfo(); // Reuse parent method
        System.out.println("Roll Number: " + rollNumber);
        System.out.println("Course: " + course);
    }
}

// --- Derived Class: GraduateStudent ---
static class GraduateStudent extends Student {
    String specialization;
    String researchTopic;

    // Constructor with super() chaining
    GraduateStudent(String name, int age, String rollNumber,
String course, String specialization, String researchTopic) {
        super(name, age, rollNumber, course);
        this.specialization = specialization;
        this.researchTopic = researchTopic;
    }

    void displayGraduateStudentInfo() {
        displayStudentInfo(); // Reuse parent method
        System.out.println("Specialization: " + specialization);
        System.out.println("Research Topic: " + researchTopic);
    }
```

```
    }

    // --- Main Method ---
    public static void main(String[] args) {
        // Creating a GraduateStudent object
        GraduateStudent gs = new GraduateStudent(
                "Aarav Sharma",
                24,
                "U2023105",
                "M.Tech Computer Science",
                "Artificial Intelligence",
                "Machine Learning Optimization"
        );

        // Display full details
        gs.displayGraduateStudentInfo();
    }
}
```

OUTPUT :

```
Name: Chetan Bhuyal
Age: 21
Roll Number: U2023105
Course: B.E Computer Science
Specialization: Artificial Intelligence
Research Topic: Machine Learning Optimization
```