**Demonstrate implementation of O/R Mapping**

O/R Mapping (Object-Relational Mapping) is the technique of mapping Java objects (entities) to database tables using annotations like @Entity, @Id, and relationship mappings like @OneToMany, @ManyToOne, @ManyToMany, etc.

**Entity Relationship Example :**

**Student.java**

@Entity

public class Student {

@Id @GeneratedValue

private Long id;

private String name;

@OneToMany(mappedBy = "student", fetch = FetchType.LAZY)

private List<Enrollment> enrollments;

// Getters and setters

}

**course.java**

@Entity

public class Course {

@Id @GeneratedValue

private Long id;

private String title;

@OneToMany(mappedBy = "course", fetch = FetchType.EAGER)

private List<Enrollment> enrollments;

@ManyToOne

@JoinColumn(name = "teacher\_id")

private Teacher teacher;

// Getters and setters

}

**Enrollment.java** (**ManyToOne on both sides)**

@Entity

public class Enrollment {

@Id @GeneratedValue

private Long id;

private LocalDate enrolledDate;

@ManyToOne

@JoinColumn(name = "student\_id")

private Student student;

@ManyToOne

@JoinColumn(name = "course\_id")

private Course course;

// Getters and setters

}

**Teacher.java (OneToMany)**

@Entity

public class Teacher {

@Id @GeneratedValue

private Long id;

private String name;

@OneToMany(mappedBy = "teacher", fetch = FetchType.LAZY)

private List<Course> courses;

// Getters and setters

}