```java
import model.Department;
import model.Teacher;
import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.Transaction;
import org.hibernate.cfg.Configuration;

import java.util.ArrayList;

public class App {
    public static void main(String[] args) {
//        manyToOne();
        oneToMany();
    }


    public static void manyToOne() {
        SessionFactory factory = new Configuration().configure().buildSessionFactory();
        Session session = factory.openSession();
        Transaction transaction = session.beginTransaction();

        Department dept1 = new Department("IT");
        Department dept2 = new Department("HR");

        Teacher t1 = new Teacher("1000", "MHaseeb", dept1);
        Teacher t2 = new Teacher("2220", "Shahparan", dept1);
        Teacher t3 = new Teacher("3000", "James", dept1);
        Teacher t4 = new Teacher("40000", "Joseph", dept2);

        session.persist(dept1);
        session.persist(dept2);

        session.persist(t1);
        session.persist(t2);
        session.persist(t3);
        session.persist(t4);
        transaction.commit();
    }

    public static void oneToMany() {
        SessionFactory factory = new Configuration().configure().buildSessionFactory();
        Session session = factory.openSession();
        Transaction t = session.beginTransaction();

        Teacher t1 = new Teacher("1000", "MHaseeb");
        Teacher t2 = new Teacher("2220", "Shahparan");
        Teacher t3 = new Teacher("3000", "James");
        Teacher t4 = new Teacher("40000", "Joseph");
        Teacher t5 = new Teacher("200", "Ali");

        ArrayList<Teacher> teacherArrayList = new ArrayList<>();
        teacherArrayList.add(t1);
        teacherArrayList.add(t2);
        teacherArrayList.add(t3);
        teacherArrayList.add(t4);
        teacherArrayList.add(t5);

        session.persist(t1);
        session.persist(t2);
        session.persist(t3);
        session.persist(t4);
        session.persist(t5);

        Department department = new Department();
        department.setDeptName("Development");
        department.setTeacherList(teacherArrayList);

        session.persist(department);
        t.commit();
    }
}

```

```java
package model;

import java.io.Serial;
import java.io.Serializable;

import jakarta.persistence.*;

@Entity
@Table
public class Teacher implements Serializable {
    @Serial
    private static final long serialVersionUID = 1L;
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private int teacherId;
    private String salary;
    private String TeacherName;
//    @ManyToOne
//    private Department department;

    public Teacher(int teacherId, String salary, String teacherName) {
        super();
        this.teacherId = teacherId;
        this.salary = salary;
        TeacherName = teacherName;
    }

    public Teacher() {
    }

    public Teacher(String salary, String teacherName, Department department){
        this.salary=salary;
        this.TeacherName=teacherName;
//        this.department=department;
    }
    public Teacher(String salary, String teacherName){
        this.salary=salary;
        this.TeacherName=teacherName;
    }

//    public Department getDepartment(){
//        return department;
//    }
    public int getTeacherId(){
        return teacherId;
    }
    public void setTeacherId(int teacherId) {
        this.teacherId = teacherId;
    }
    public String getSalary() {
        return salary;
    }
    public void setSalary(String salary) {
        this.salary = salary;
    }
    public String getTeacherName() {
        return TeacherName;
    }
    public void setTeacherName(String teacherName) {
        TeacherName = teacherName; }


}

```

```java
package model;
import jakarta.persistence.*;
import java.io.Serial;
import java.io.Serializable;
import java.util.List;

@Entity
@Table
public class Department implements Serializable {
    @Serial
    private static final long serialVersionUID=1L;

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private int deptId;
    private String deptName;
    @OneToMany(targetEntity= Teacher.class, cascade = {CascadeType.ALL})
    private List<Teacher> teacherList;
    public Department(int deptId, String deptName){
        super();
        this.deptId=deptId;
        this.deptName=deptName;
    }

    public List<Teacher> getTeacherList() {
        return teacherList;
    }
    public void setTeacherList(List<Teacher> teacherList) {
        this.teacherList = teacherList;
    }


    public Department(){}

    public Department(String deptName){
        this.deptName=deptName;
    }

    public int getDeptId(){
        return deptId;
    }

    public void setDeptId(int deptId) {
        this.deptId = deptId;
    }

    public String getDeptName() {
        return deptName;
    }

    public void setDeptName(String deptName) {
        this.deptName = deptName;
    }
}

```

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE hibernate-configuration PUBLIC
        "-//Hibernate/Hibernate Configuration DTD 3.0//EN"
        "http://www.hibernate.org/dtd/hibernate-configuration-3.0.dtd">
<hibernate-configuration>
    <session-factory>
        <!-- Drop and re-create the database on startup -->
        <property name="hibernate.hbm2ddl.auto">create-drop </property>

        <!-- Database connection settings -->
        <property name="connection.driver_class">com.mysql.cj.jdbc.Driver</property>
        <property name="connection.url">jdbc:mysql://localhost:3306/usersDb</property>
        <property name="connection.username">root</property>
        <property name="connection.password">SzhengSQL123</property>

        <!-- MySQL DB dialect -->
        <property name="dialect">org.hibernate.dialect.MySQLDialect</property>
        <!-- print all executed SQL on console -->
        <property name="hibernate.show_sql" >true </property>
        <property name="hibernate.format_sql" >true </property>

<!--        &lt;!&ndash;   Mapping entity file →-->
        <mapping class="model.Teacher"/>
        <mapping class="model.Department"/>
    </session-factory>
 </hibernate-configuration>


```