```java
import model.Address;
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();
    oneToOne();
    }



    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);

```

```
36          session.persist(t1);
37          session.persist(t2);
38          session.persist(t3);
39          session.persist(t4);
40          transaction.commit();
41      }
42
43      public static void oneToMany() {
44          SessionFactory factory = new Configuration().
    configure().buildSessionFactory();
45          Session session = factory.openSession();
46          Transaction t = session.beginTransaction();
47
48          Teacher t1 = new Teacher("1000", "MHaseeb");
49          Teacher t2 = new Teacher("2220", "Shahparan"
    );
50          Teacher t3 = new Teacher("3000", "James");
51          Teacher t4 = new Teacher("40000", "Joseph");
52          Teacher t5 = new Teacher("200", "Ali");
53
54          ArrayList<Teacher> teacherArrayList = new
    ArrayList<>();
55          teacherArrayList.add(t1);
56          teacherArrayList.add(t2);
57          teacherArrayList.add(t3);
58          teacherArrayList.add(t4);
59          teacherArrayList.add(t5);
60
61          session.persist(t1);
62          session.persist(t2);
63          session.persist(t3);
64          session.persist(t4);
65          session.persist(t5);
66
67          Department department = new Department();
68          department.setDeptName("Development");
69          department.setTeacherList(teacherArrayList);
70
71          session.persist(department);
72          t.commit();
73      }
```

```
74
75      public static void oneToOne(){
76          System.out.println("Maven + Hibernate + SQL
    One to One Mapping Annotations");
77
78          SessionFactory factory = new Configuration
    ().configure().buildSessionFactory();
79          Session session = factory.openSession();
80          Transaction t = session.beginTransaction();
81          Address a1 = new Address("27th street", "NYC
    ", "NY", 11103);
82          Address a2 = new Address("28th street", "
    Buffalo", "NY", 15803);
83
84          Teacher t1 = new Teacher("1000", "MHaseeb");
85          Teacher t2 = new Teacher("2220", "Shahparan"
    );
86          t1.setAddress(a1);
87          t2.setAddress(a2);
88
89          session.persist(a1);
90          session.persist(t1);
91          session.persist(a2);
92          session.persist(t2);
93
94          t.commit();
95      }
96 }
97
```

```java
package model;

import jakarta.persistence.*;

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

@Entity
@Table
public class Address implements Serializable {
    @Serial
    private static final long serialVersionUID = 1L;
    @Id
    @GeneratedValue(strategy = GenerationType.
IDENTITY)
    private int addressId;
    private String street;
    private String city;
    private String state;
    private int zipCode;
    public Address(){}

    public Address(String street, String city, String
 state, int zipCode) {
        this.street = street;
        this.city = city;
        this.state = state;
        this.zipCode = zipCode;
    }

    public int getAddressId() {
        return addressId;
    }

    public void setAddressId(int addressId) {
        this.addressId = addressId;
    }

    public String getStreet() {
        return street;
    }
```

```
40
41      public void setStreet(String street) {
42          this.street = street;
43      }
44
45      public String getCity() {
46          return city;
47      }
48
49      public void setCity(String city) {
50          this.city = city;
51      }
52
53      public String getState() {
54          return state;
55      }
56
57      public void setState(String state) {
58          this.state = state;
59      }
60
61      public int getZipCode() {
62          return zipCode;
63      }
64
65      public void setZipCode(int zipCode) {
66          this.zipCode = zipCode;
67      }
68  }
69
```

```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;

    @OneToOne(cascade = CascadeType.ALL)
    private Address address;
    public Address getAddress(){
        return address;
    }

    public void setAddress(Address address) {
        this.address = address;
    }

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

    public Teacher() {
    }
```

```
40
41      public Teacher(String salary, String teacherName
    , Department department){
42          this.salary=salary;
43          this.TeacherName=teacherName;
44  //          this.department=department;
45      }
46      public Teacher(String salary, String teacherName
    ){
47          this.salary=salary;
48          this.TeacherName=teacherName;
49      }
50
51  //    public Department getDepartment(){
52  //        return department;
53  //    }
54      public int getTeacherId(){
55          return teacherId;
56      }
57      public void setTeacherId(int teacherId) {
58          this.teacherId = teacherId;
59      }
60      public String getSalary() {
61          return salary;
62      }
63      public void setSalary(String salary) {
64          this.salary = salary;
65      }
66      public String getTeacherName() {
67          return TeacherName;
68      }
69      public void setTeacherName(String teacherName) {
70          TeacherName = teacherName; }
71
72
73  }
74
```

```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;
    }

```

```
39      public int getDeptId(){
40          return deptId;
41      }
42
43      public void setDeptId(int deptId) {
44          this.deptId = deptId;
45      }
46
47      public String getDeptName() {
48          return deptName;
49      }
50
51      public void setDeptName(String deptName) {
52          this.deptName = deptName;
53      }
54  }
55
```

```
 1 <?xml version="1.0" encoding="UTF-8"?>
 2 <!DOCTYPE hibernate-configuration PUBLIC
 3         "-//Hibernate/Hibernate Configuration DTD 3.0
   //EN"
 4         "http://www.hibernate.org/dtd/hibernate-
   configuration-3.0.dtd">
 5 <hibernate-configuration>
 6     <session-factory>
 7         <!-- Drop and re-create the database on
   startup -->
 8         <property name="hibernate.hbm2ddl.auto">
   create-drop </property>
 9
10         <!-- Database connection settings -->
11         <property name="connection.driver_class">com.
   mysql.cj.jdbc.Driver</property>
12         <property name="connection.url">jdbc:mysql://
   localhost:3306/usersDb</property>
13         <property name="connection.username">root</
   property>
14         <property name="connection.password">
   SzhengSQL123</property>
15
16         <!-- MySQL DB dialect -->
17         <property name="dialect">org.hibernate.
   dialect.MySQLDialect</property>
18         <!-- print all executed SQL on console -->
19         <property name="hibernate.show_sql" >true </
   property>
20         <property name="hibernate.format_sql" >true
   </property>
21
22 <!--         &lt;!&ndash;   Mapping entity file →-->
23         <mapping class="model.Teacher"/>
24         <mapping class="model.Department"/>
25         <mapping class="model.Address"/>
26     </session-factory>
27  </hibernate-configuration>
28
29
```