

Screenshots: attached to readme

## Address.java

```
@Entity
@Getter
@Setter
@NoArgsConstructor
public class Address {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private int aId;
    private String area;
    private String district;
    private int pincode;
    private int employeeId;

    public Address(String area, String district, int pincode) {
        this.area = area;
        this.district = district;
        this.pincode = pincode;
    }
}
```

## Courses.java

```
@Entity
@Getter
@Setter
@NoArgsConstructor
@ToString
public class Courses {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private int courseId;
    private String courseName;

    @ManyToMany(mappedBy="coursesSet")
    private Set<Employee> employeeSet;
}
```

## Employee.java

```
@Entity
@Getter
@Setter
@NoArgsConstructor
@ToString
public class Employee {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private int empId;
    private String firstName;
    private String lastName;

    @OneToMany(mappedBy = "employeeId", fetch = FetchType.EAGER, cascade = CascadeType.ALL)
    private Set<Address> addresses;

    @ManyToMany(cascade=CascadeType.ALL)
    @JoinTable(name="EmployeeCourse", joinColumns={@JoinColumn(referencedColumnName="empId")},
        inverseJoinColumns={@JoinColumn(referencedColumnName="courseId")})
    private Set<Courses> coursesSet;
}
```

```

    public Employee(String firstName, String lastName) {
        this.firstName = firstName;
        this.lastName = lastName;
    }
}

```

## PersonalDetails.java

```

@Entity
@Getter
@Setter
@NoArgsConstructor
public class PersonalDetails implements Serializable {
    private String email;
    private String mobile;

    @Id
    @OneToOne
    @MapsId
    private Employee employee;

    public PersonalDetails(String email, String mobile, Employee employee) {
        this.email = email;
        this.mobile = mobile;
        this.employee = employee;
    }
}

```

## MainClass

```

public class EmployeeDemo {
    public static void main(String[] args) throws Exception{
        SessionFactory sessionFactory = new Configuration()
            .configure("hibernate.cfg.xml")
            .addAnnotatedClass(Employee.class)
            .addAnnotatedClass(Courses.class)
            .addAnnotatedClass(Address.class)
            .addAnnotatedClass(PersonalDetails.class)
            .buildSessionFactory();
        Session session = sessionFactory.openSession();

        Employee employee1 = new Employee("kalai", "chelvan");
        Employee employee2 = new Employee("kannan", "m");
        Set<Employee> employees = new HashSet<>();
        employees.add(employee1);
        employees.add(employee2);

        PersonalDetails personalDetails1 = new PersonalDetails("k@gmail.com", "992218",
employee1);
        PersonalDetails personalDetails2 = new PersonalDetails("chelvan@gmail.com", "827399",
employee2);

        Address address1 = new Address("T Nagar", "Chennai", 600025, employee1);
        Address address2 = new Address("Thiruvandipuram", "Cuddalore", 607401, employee1);

        Set<Address> employee1Address = new HashSet<>();
        employee1Address.add(address1);
        employee1Address.add(address2);
        employee1.setAddresses(employee1Address);

        Courses courses1 = new Courses();
        Courses courses2 = new Courses();
        courses1.setCourseName("Spring MVC Afternoon");
        courses2.setCourseName("Spring Core Forenoon");
        Set<Courses> courses = new HashSet<>();
        courses.add(courses1);
        courses.add(courses2);

        // ManyToMany
    }
}

```

```
employee1.setCoursesSet(courses);
employee2.setCoursesSet(courses);
courses1.setEmployeeSet(employees);
courses2.setEmployeeSet(employees);

try{
    session.beginTransaction();
    session.save(courses1);
    session.save(courses2);
    session.save(employee1);
    session.save(employee2);
    session.save(personalDetails1);
    session.save(personalDetails2);
    session.getTransaction().commit();
}finally {
    session.close();
    sessionFactory.close();
}
}
```