<table>
<tr><td>@Entity</td><td>Convert class into entity, will create table by class name, to change table name use @Table</td></tr>
<tr><td>@Table</td><td>It will help to change the table name</td></tr>
<tr><td>@Id</td><td>It will treat the member as primary key in the table, If the value should be generated automatically then use @GeneratedValue</td></tr>
<tr><td>@GeneratedValue</td><td>It will select the best possible way to generate primary key value, if you want to change it then use strategy attribute
@GeneratedValue(strategy=GenerationType.IDENTITY)</td></tr>
<tr><td>@Column</td><td>@Column(name="name",nullable=false,unique=true,length=30)
private String pname;</td></tr>
<tr><td>@Embeddable</td><td>If there are 2 classes and for both classes you want single table then use @Embeddable for one class</td></tr>
<tr><td>@Embbed</td><td>If there are 2 classes and for both classes, you want single table then use @Embeded In the class in which you are adding another class object as member
Both user and address members will get saved in same table
<table>
<tr><td>user</td><td>Address</td></tr>
<tr><td>@Entity
Class User{

@Embedded
Address addr

}</td><td>@Embbedable
Class Address{

}</td></tr>
<tr><td></td><td></td></tr>
</table>
</td></tr>
<tr><td>@OneToOne</td><td>This annotation will add primary key of another class as foreign key in the current table
It is unidirectional one to one mapping, it will add address id as foreign key in user table
<table>
<tr><td>user</td><td>Address</td></tr>
<tr><td>@Entity
Class User{

@OneToOne
Address addr

}</td><td>@Entity
Class Address{
@Id
private int addrid

}</td></tr>
</table>
</td></tr>
</table>

Bidirectional OneToOne

It is bidirectional one to one mapping, it will add address id as foreign key in user table

and, it will add user id as foreign key in address table

To avoid adding foreign key in both tables use mapped by, in the address class, so foreign key will not get added in address class table

user	Address

@Entity
Class User{
 	
@OneToOne
Address addr
 	
}
 	
	@Entity
	Class Address{
	@Id
	  private int addrid
	@OneToOne
	User u;
	
	}

Using mappedBy

user	Address

@Entity
Class User{
 	
@OneToOne
Address addr
 	
}
 	
	@Entity
	Class Address{
	@Id
	  private int addrid
	@OneToOne(mappedBy=”addr”
	User u;
	
	}

To avoid join query on both address and user table use

Fetch=FetchType.lazy

Session.get method is by default eager fetching

Session.load method is by default lazy fetching

Cascade=casecade.all

Will be helpful to save address automatically when you save user object, or delete address object if you delete user object

Assignment1:

Create relation between course and Faculty class using one to one biderection mapping