\*\*\* JBK Hibernate \*\*\*

­­­­­­­­­­­­­­­­­**Four types Create a Pojo class / Persist class /Encapsulated Class/Entity class/model class**

1. **Default Constructor**
2. **Parameter Constructor**
3. **Getter and Setter**
4. **toString method**

**@Entity Annotation**

-The Enterprise JavaBased 3 standard annotations are contained in the javax.persistence package, so we import this package as the first step. Second, we used the @Entity annotation to the Student class, which marks this class as an entity bean, so it must have a no-argument constructor that is visible with at least protected scope.

**@Id Annotation**

-Each entity bean will have a primary key, which you annotate on the class with the @Id annotation. The primary key can be a single field or a combination of multiple fields depending on your table structure.

**Test Main Class**

**1) Configuration**

-Hibernate also requires a set of configuration settings related to database and other related parameters. All such information is usually supplied as a standard Java properties file called hibernate.properties, or as an XML file named hibernate.cfg.xml.

**2) Annotated Class Example**

-As I mentioned above while working with Hibernate Annotation, all the metadata is clubbed into the POJO java file along with the code, this helps the user to understand the table structure and POJO simultaneously during the development.

**3) configure()**

- Configure class method

**4) SessionFactory**

-A sessionFactory object represents a single database or datastore. It is a thread safe object. It is built only once at the start of an application.

**5) Session**

-A Session is used to get a physical connection with a database. The Session object is lightweight and designed to be instantiated each time an interaction is needed with the database. Persistent objects are saved and retrieved through a Session object.

**6) Transaction**

-Transaction represents a single unit of work.

**7) void commit()**

-commits the transaction Permanent data Storage in dataBase.

**8) void rollback()**

-cancels the transaction.

**9) hibernate.cfg.xml**

Generally, the hibernate.cfg.xml file contains the database connectivity and Enitity classes information.