

# Hibernate Framework 3.5

---

**Duration:** 2 Days (hands-on)

**Prerequisites:** Participants should be comfortable with JEE application development

Following is a day-wise breakup of the different topics to be covered during the training program

## Day 1

- Introduction to Hibernate
- Understanding ORM (Object Relational Mapping)
- Understanding Transient, Persistent and Detached Object states
- Issues while writing manual JDBC code
- Dialect classes in Hibernate
- Hibernate and JPA (Java Persistence API)
- POJO approach and its benefits
- Writing persistence classes
- Writing database mapping xml (.hbm.xml)
- Understanding SessionFactory and Session API
- Handling CRUD operations in Hibernate
- Handling Transient, Persistent and Detached Object states
- Understanding xml configuration elements like *id*, *composite-id*, *natural-id*, *dynamic-update*, *mutable*, *lazy*, etc...
- Mapping Blob, Clob in Hibernate
- Introducing inheritance
  - Mapping Inheritance between classes with tables in database
  - Understanding xml elements like *subclass*, *joined-subclass*, etc...
  - Using xml element called as *any*
- Introducing associations
  - one-to-one, one-to-many, many-to-many
  - Exploring how to handle the same in Hibernate
  - Understanding xml elements like *one-to-one*, *one-to-many*, *many-to-many*, *set*, *list*, *cascade*, *inverse*, *key*, etc...
- Mixing inheritance and associations

## Day 2

- Understanding issues while fetching data from database
- Exploring fetching strategies supported by Hibernate
  - Lazy fetching
  - Eager fetching

- Join fetching
- Batch fetching
- Subselect fetching
- Immediate fetching
- Introducing HQL (Hibernate Query Language)
  - Need for HQL
  - Writing simple queries with HQL
  - Handling inner joins, outer joins, self joins with HQL
  - Using HQL for insert, update and delete also
- Introducing Criteria API
  - Need for writing dynamic queries
  - Exploring Criteria API for generating dynamic SQL
- Concurrency support in Hibernate
  - Issues while concurrent updates
  - Understanding locking and its types
    - Pessimistic locking
    - Optimistic locking
  - Understanding how Hibernate supports locking
  - Using version / timestamp for optimistic locking
  - Using pessimistic locking
- Caching
  - Understanding First Level Cache and Second Level Cache
  - Need for Second Level Cache
  - Using EhCache as the Second Level Cache provider
  - Understanding how caching improves application performance
  - Configuring EhCache