

## Hibernate Course Content:35-40hours

### Course Outline

#### 1. Understanding object/relational persistence

- What is persistence? Relational databases
- Persistence in object-oriented applications The paradigm mismatch
- The problem of granularity
- The problem of subtypes
- The problem of identity Problems relating to associations
- The problem of object graph navigation Persistence layers and alternatives Object/relational mapping
- What is ORM?
- Generic ORM problems
- Why ORM?

#### 2. Introducing and integrating Hibernate

- "Sample Application" with Hibernate Understanding the architecture
- The core interfaces Callback interfaces Basic configuration
- Creating a Session
- Factory Configuration in non-managed environments Configuration
- Advanced configuration settings Using XML-based configuration

#### 3. Mapping

- JNDI-bound SessionFactory Logging persistent classes
- Implementing the domain model Addressing leakage of concerns Transparent and automated persistence Writing POJOs
- Implementing POJO associations Adding logic to accessor methods Defining the mapping metadata Metadata in XML
- Basic property and class mappings Attribute-oriented programming Manipulating metadata at runtime Understanding object identity

#### 4. Working

- Identity versus equality
- Database identity with Hibernate Choosing primary keys
- Fine-grained object models
- Entity and value types Using components
- Mapping class inheritance Table per concrete class Table per class hierarchy Table per subclass
- Choosing a strategy
- Introducing associations
- Managed associations
- The simple association
- Making the association bidirectional Parent & Child relationship with persistent objects
- The persistence lifecycle
- Transient objects Persistent objects

- Detached objects
- The scope of object identity Outside the identity scope Implementing equals() and hashCode()
- The persistence manager Making an object persistent
- Updating the persistent state of a detached instance
- Retrieving a persistent object
- Updating a persistent object
- Making a persistent object transient
- Making a detached object transient
- Using transitive persistence in Hibernate Cascading persistence with Hibernate
- Distinguishing between transient and detached instances
- Retrieving objects
- Retrieving objects by identifier
- Introducing HQL Query by criteria Fetching strategies
- Selecting a fetching strategy in mappings

#### **5. Transactions, concurrency, and caching Transactions, concurrency, and caching**

- Understanding database transactions JDBC and JTA transactions
- The Hibernate Transaction API
- Flushing the Session
- Understanding isolation levels
- Choosing an isolation level
- Setting an isolation level
- Using pessimistic locking
- Working with application transactions
- Implement optimistic locking
- Caching theory and practice
- Caching strategies and scopes
- The Hibernate cache architecture

#### **6. Advanced mapping concepts**

- Built-in mapping types
- Using mapping types
- Mapping collections of value types
- Sets, bags, lists, and maps
- Mapping entity associations
- One-to-one associations
- Many-to-many associations
- Mapping polymorphic associations
- Polymorphic many-to-one associations
- Polymorphic collections
- Polymorphic associations and table-per- concrete-class

#### **7. Retrieving objects efficiently**

- Executing queries
- The query interfaces

- Binding parameters
- Using named queries
- Basic queries for objects
- The simplest query Using aliases
- Polymorphic queries Restriction
- Joining associations
- Hibernate join options
- Fetching associations using aliases with joins 2
- Using implicit joins
- Theta-style joins
- Comparing identifiers
- Writing report queries
- Projection
- Using aggregation Grouping
- Restricting groups with having
- Improving performance with report queries
- Advanced query techniques
- Dynamic queries Collection filters
- Subqueries, Native SQL queries

#### **8. Writing Hibernate applications**

- Designing layered applications
- Using Hibernate in a servlet engine
- Using Hibernate in an EJB container
- Implementing application transactions
- Using detached persistent objects 324 - Using a long session 325 - Choosing an approach to application transactions 329
- Handling special kinds of data 330
- Legacy schemas and composite keys
- Audit logging

#### **9. Using the toolset**

- Development processes
- Top down & Bottom up
- Automatic schema generation
- Preparing the mapping metadata
- Creating the schema
- Updating the schema
- Generating POJO code
- Adding meta-attributes
- Generating finders Configuring hbm2java
- Running hbm 2 Java 3
- Restricting tables and relationships
- Customizing the metadata generation