

# Hibernate

**Course No.**

3459

**Description**

Hibernate is the popular open source persistence framework that adds Object Relational Mapping to your applications. This course will both introduce Hibernate and allow you to explore some of its most advanced features including the use of object caching which can greatly improve application performance. You will learn how to replace complex JDBC code with simple Hibernate calls to provide access to a relational database without the difficulty and overhead of using EJB entity beans. As Hibernate usage has increased, design patterns have been identified which will allow you to take advantage of best practices. The course will highlight these using real world examples. And finally, this course will explore the similarities between Hibernate and the emerging new EJB 3.0 standard.

**Audience**

Developers building Java applications will require connection to a relational database.

**Objectives**

- Knowledge of the Java Programming Language
- Familiarity with SQL and Relational Databases

**Major Topics**

- Introduction
- Hibernate Overview
- Configuration and Session Factory
- Session
- Mapping Associations
- Advanced Mappings
- Query and Criteria
- Fetching Strategies and Performance
- Caching (Using EhCache)
- Patterns and Best Practices
- Transactions
- Tools

**Duration**

3 days



## Course Contents

### 1. Introduction

- Classroom logistics
- Course overview
- Tools and software discussion

### 2. Hibernate Overview

- Object Relational mismatch
- What is an ORM?
- Alternatives to Hibernate
  - EJB
  - iBATIS
  - JDBC
- Hibernate architecture overview
  - Introduction to Hibernate classes

### 3. Configuration and Session Factory

- Discussion of Configuration object
- Discussion of Session Factory object
- Configuration properties
  - JDBC properties
  - Other properties
- Building a Configuration
  - Programmatically
  - Using .properties file
  - Using XML configuration file
  - Combining build methods
- Reusing the Session Factory
  - Threadlocal variable pattern

### 4. Session

- Persistence management
- HelloWorld example
  - Simple Employee mapping
- CRUD examples
  - insert
  - get vs. load
  - update
  - delete
- Review of Javadoc for Session
- Object States
  - Review of State Transition diagram
  - Transient state
  - Persistent state

- Detached state
- Maintaining object identity
- Re-attaching detached objects

### 5. Mapping Associations

- Entity types
- Value types
- Components
- Hibernate Types
- Custom value types

### 6. Advanced Mappings

- Subclasses and inheritance
- Other mappings

### 7. Query and Criteria

- Hibernate Query Language (HQL)
- WHERE clause
- Sorting
- Projections
- Criteria queries

### 8. Fetching Strategies and Performance

- Review of Hibernate fetching
- Strategies
  - JOIN, SUBSELECT, BATCH
- Lazy initialization
- Transitive persistence
- Cascade feature
- Impact of fetching strategies

### 9. Caching (Using EhCache)

- Cache concepts
- First Level Cache
- Second Level Cache
  - Cache Modes
  - Using statistics to monitor cache
  - EhCache Configuration

### 10. Patterns and Best Practices

- Threadlocal pattern
- Save Session in View
- DAO Pattern



## 11. Transactions

- JDBC and JTA transactions
- Hibernate transactions
- Long transactions
- Transaction demarcation
- Handling transaction failure
- Optimistic and pessimistic locking
- Versioning

## 12. Tools

- POJO generation (hbm2ddl)
- Schema generation (hbm2java)
- Ant Tasks
- Eclipse plugins

## Optional Topics

### 13. EJB 3.0

- JDK 1.5 Annotations

