

---

## Using Hibernate in Java Applications

<b>Days</b>	3
<b>Course code</b>	QAJAVHIB

---

### Overview

Hibernate is the most popular object-relational mapping tool, which uses a configuration to persist objects in an RDBMS. Advanced features of Hibernate include transactions and caching.

This highly practical course covers each of this technology, and demonstrates how it can be used to produce a robust and flexible multi-tier enterprise system.

### Presentation Method

Classroom presentation with hands-on practical exercises, using Oracle Express 10g, MySql 5.0.15 and developed with the Eclipse 3.2 editor, using Java 5 or 6.

### Prerequisites

- 
- This advanced course is designed for experienced Java developers.
  - Strong Java skills
  - Some exposure to large-scale transactional systems is beneficial
  - Working knowledge of modelling, perhaps using UML
- 

### Delegates will learn how to

- 
- Use the Hibernate Framework to access data
  - Understand when to use an ORM layer
  - Map complex object models to databases
  - Manage transactions using Hibernate
  - Use caching to improve database performance
  - Query objects using the Hibernate Query Language
  - Make use of Hibernate's advanced features
  - Use Hibernate with JEE
- 

### Course outline

#### Introduction to Hibernate

- 
- The Hibernate Framework
  - Why have ORM's?
  - Hibernate principles
  - Configuration

---

## **Persistent Objects**

- Object lifecycles
- Transient objects
- Flushing
- The role of the cache

---

## **Mapping**

- How to map objects
- Single value objects
- Multi-value objects
- Relationships and Inheritance mapping options
- Using annotations to map

---

## **Hibernate Query Language**

- Why another query language?
- Syntax and usage
- Deciding on the caching strategy

---

## **Java Persistence API**

- JPA Lifecycle
- The Entity Manager
- JPA Query Language

---

## **Concurrency**

- Concurrency options
- Pessimistic locking
- Optimistic locking

---

## **Transactions**

- Configuring Transaction support
- Session and Transaction Scope

---

## **Architecture**

- Data Access Objects
- Controllers
- Enterprise JavaBeans Integration

---

## **Legacy Systems Integration**

- Key management

- 
- Stored Procedures and Triggers
  - Complex database schemas

---

### **Hibernate Performance techniques**

---

- Lazy Loading
- The second level cache
- Optimizing collection performance
- Propriety SQL and dialects