

Hibernate is an open source object/relational (OR) persistence and query service for Java. Hibernate lets you develop persistent classes following common Java idioms - including association, inheritance, polymorphism, composition and the Java collections framework.

This course covers everything you need to know to begin working with Hibernate in a very short time. It covers all the important concepts necessary to access and update data stored in relational databases. It includes an extensive series of labs to exercise all major capabilities.

Course Objectives:

- Understand the benefits of Hibernate.
- Understand the Hibernate architecture.
- Create Hibernate-based applications.
- Understand and use Hibernate mapping to map persistent objects to the database.
- Understand and work with collections and associations.
- Utilize Hibernate’s versioning support.
- Map inheritance hierarchies using Hibernate.
- Work with Hibernate queries, HQL, and Criteria.
- Performance tune your Hibernate applications.
- Understand Hibernate transaction support.
- Understand the relationship between Hibernate and Java Persistence/EJB 3.
- Use the new Hibernate annotations to do OR mapping.

Audience: Individuals needing to access relational databases from Java programs.

Prerequisites: Intermediate knowledge of Java programming and a basic understanding of relational databases are recommended.

Number of Days: 3 days

<p>1. Hibernate Overview The Issues with Persistence Layers Object-Relational Mapping (ORM) Issues Issues with JDBC Alone Hibernate Overview Hibernate Benefits Hibernate Environments Hibernate Architecture More Detailed Architecture</p>	<p>3. Mapping a Simple Class Persistent Entity Classes Persistent Classes The Event Class The id Property The Hibernate Mapping File</p>
<p>2. Using Hibernate Acquiring Hibernate Using Hibernate Configuring Hibernate hibernate.cfg.xml Elements</p>	<p>SessionFactory Configuration SessionFactory Configuration Properties The Configuration Class The SessionFactory Interface SessionFactory API The Session Interface Sessions and Transactions</p>

- The <hibernate-mapping> Element
- The <class> Element
- The EVENTS Table
- Mapping the id Property with <id>
- More About Primary Keys
- Generating the id Value
- Mapping Properties with <property>
- Hibernate Mapping Types
- Common Hibernate Type Mappings
- Filed Access or Property Access
- The Mapping File
- Hibernate Sessions
- The Session Interface
- Retrieving Persistent Objects
- 4. Logging**
 - Hibernate.show_sql
 - Apache Log4J
 - Hibernate log4j.properties File
 - The log4j.properties File
 - Modifying log4j.properties for Hibernate
 - Hibernate Logging Categories
- 5. Appendix – Log4J**
 - Apache Log4J
 - log4j Loggers
 - Logger Hierarchy
 - Logger Levels
 - Appenders
 - Aprender Additivity
 - Layout
 - log4j Configuration File
 - PatternLayout
 - Multiple Layouts
 - Some log4j Appenders
- 6. Inserting and Updating**
 - Inserting Instances
 - Modifying a Persistent Instance
 - Deleting an Instance
- 7. Querying and Hibernate Query Language (HQL)**
 - Hibernate Query Language
 - HQL Basics
 - Executing a Query
 - Other Common Query Methods
 - Where Clause/Restriction
 - HQL Operators and Expressions
 - Query Parameters
- Using Query Parameters
- Named Queries
- Projection Queries
- Projection Queries Returning Tuples
- Additional Query Capabilities
- Aggregate Functions
- 8. Transaction Definition**
 - Transaction Lifecycle
 - Transactions Modularize Systems
 - Transactions Clarify Systems
- 9. Hibernate and Transactions**
 - Hibernate and Transactions
 - Hibernate Transaction Demarcation
 - Working with Transactions
 - The Hibernate Transaction API
 - Working in a Managed Environment
- 10. The Persistence Lifecycle**
 - Hibernate Object States
 - Transient and Persistent State
 - Detached and Removed State
 - Hibernate Object States and Transitions
 - The Persistence Context
 - Session/Persistence Context Lifespan
 - Session-per-Request
 - Session Propagation
 - First – Acquiring a SessionFactory Instance
 - Contextual Session
 - Using Contextual Sessions
 - What is the “Current” Context
 - Contextual Session Scope
 - The Persistence Context as Cache
 - Synchronization to the Database
 - Flushing the Session
 - Persistence Context and Object Identity
 - Yes, It’s Complicated
- 11. Versioning and Optimistic Locking**

- Detached Objects and Optimistic Locking
- Using a Detached Instance
- Optimistic Locking and Versioning
- Version Property in Java Class
- Version Element in Mapping File
- Automatic Version Maintenance
- Updating a Detached Instance
- session.saveOrUpdate()
- The unsaved-value Attribute
- Locking Objects
- Lock Modes
- 12. Relationships Overview**
 - Object Relationships
 - Characteristics of Relationships
 - Directionality
 - Characteristics of Relationships
- 13. Collections of Value Objects**
 - Collections of Values
 - Modeling a Set of Values
 - Mapping the Set of Values
 - Using a Set of Values
 - More on the Java Collection Type
 - Using the Java Collection Types
 - Modeling a List of Values
 - Mapping a List of Values
 - Sorted and Ordered Collections
 - Collections of Components
 - Mapping Collections of Components
- 14. Mapping Entity Relationships**
 - Unidirectional Many-To-One Relationship
 - The Table Structure – Many-To-One
 - Mapping the Relationship
 - Using the Relationship
 - Bidirectional One-To-Many Relationship
 - Defining the One-To-Many Relationship
 - Mapping the One-To-Many Relationship
 - More on the Inverse Side
 - Cascading Operations
 - Transitive Persistence
 - The Cascade Attribute
 - Cascade Choices
 - Choosing Cascade Options
- Bidirectional One-To-One Relationship
- Mapping a One-To-One Relationship
- Many-To-Many Relationship
- Defining Many-To-Many Relationship
- Mapping Many-To-Many Relationship
- Lazy and Eager Loading
- Queries Across Relationships
- OUTER and FETCH JOIN
- 15. Mapping Inheritance**
 - Inheritance
 - Entity Inheritance
 - Details of Entity Inheritance
 - Single-Table Strategy
 - Class Definitions for Single-Table
 - Mapping for Single-Table
 - Single-Table: Pros and Cons
 - Table per Subclass (Joined Subclass)
 - Mapping for Table per Subclass
 - Joined: Pros and Cons
 - Table per Concrete Class
- 16. More on Querying**
 - Projection Queries
 - Aggregate Queries
 - Bulk Update and Delete
 - Executing Bulk Operations
 - Native SQL Queries
 - Refining SQL Queries
 - Retrieving Entities with SQL Queries
- 17. Filters**
 - Hibernate Filters
 - Defining and Attaching Filters
 - Using a Filter
 - Mapping a Filter to a Set
 - Collection Filters
- 18. Criteria**
 - Restrictions – Narrowing the Result Set
 - Restrictions Methods
 - Navigating Associations

- Eager Fetching
- Query by Example
- Refining the Example
- Additional Capabilities
- 19. JPA Overview**
 - Java Persistence API Overview
 - Java Persistence Environments
 - Hibernate and JPA
- 20. Mapping a Simple Class**
 - Entity Classes
 - Event Entity Mapped with JPA
 - The Entity Declaration
 - The Event Class
 - The id Property
 - Mapping Properties
 - Basic Mapping Types
- 21. Entity Manager and Persistence Context**
 - The Entity Manager & Persistence Context
 - Persistence Unit
 - persistence.xml
 - Acquiring an EntityManager
 - Working with Transactions
 - Retrieving Persistent Objects
- 22. Inserts and Queries**
 - Persisting a New Entity
 - Java Persistence Query Language
 - Executing a Query
 - WHERE Clause and Query Parameters
 - Named Queries
 - Version Property in Java Class
 - Versioned Class and Detached Objects
- 23. Relationships**
 - JPA Support for Relationships
 - Mapping the Many-To-One Relationship
 - Mapping the One-To-Many Relationship
 - Loading and Cascading
 - Queries Across Relationships
 - Inheritance
 - Entity Definitions for Single-Table
 - Entity Definitions for Joined
- 24. Components and Multi-Table Mapping**
 - Component Overview
 - Mapping a Component
- 25. Multi-Table Mapping equals() and hashCode()**
 - Defining equals() and hashCode()
 - Redefining equals()
- 26. Caching**
 - Second-Level Cache
 - Data Appropriate for Caching
 - Cache Providers
 - Configuring Caching
 - Concurrency Strategies
 - Managing the Caches
- 27. Design Considerations**
 - Long Conversations
 - Session-per-Conversation
 - Problems with Web Applications
 - Open Session in View Pattern
 - Query Efficiency Techniques
 - Beware of N+1 Select Issue
 - Prefetching Data in Batches
 - Data Access Object (DAO)
- 28. Hibernate Toolset**
 - Hibernate Tools Overview
 - Important Note on Versions
 - Optional – Hibernate Tools
 - Install Hibernate Tools
 - Hibernate Console Configuration
 - Hibernate Console Perspective
 - Hibernate Configuration View
 - Class Diagram
 - HQL Editor
 - Query Results
 - Properties View
 - SQL Preview
 - Other Capabilities