

Hibernate Essentials

The Hibernate Essentials course is for individuals who must become competent with the Hibernate or the Java Persistence API. The primary audience is intended to be Java developers who work with SQL-based database systems or database developers who are looking for an introduction to object-oriented software development. Database administrators who are interested in how ORM may affect performance and how to tune the performance of the SQL database management system and persistence layer will also find this course of value.

This course covers the foundational API of the Hibernate product and the JBoss, Inc. implementation of the JSR-220 sub-specification for Java Persistence.

Course Duration: This is a 3 day course

Course Format: This class is 50% theory and 50% hands-on lab exercises

Course Prerequisites

The prerequisite skills for this class are the following:

- An understanding of the relational persistence model
- Competency with the Java language
- Knowledge of OOAD concepts
- Familiarity with the UML
- Experience with a dialect of SQL.
- Using the JDK and creating the necessary environment for compilation and execution of a Java executable from the command line.
- An understanding of JDBC
- Completion of JBoss for Advanced J2EE Developers

Course Modules

The Hibernate Essentials course is recommended for Java developers who must become competent with a Hibernate or a Java Persistence API object/relational persistence and query service implementation. Both implementations provide powerful APIs to develop persistent classes following the object-oriented idiom. Both implementations support the use of association, inheritance, polymorphism, composition and collections with persistent classes. To fully utilize persistent class implementations, the student will also learn how to create and execute

queries using EJB-QL, the portable Hibernate SQL extension (HQL), native SQL and the object-oriented Criteria and Example API.

1. **Understanding object/relational persistence.** In this module persistent data management in the context of object-oriented applications is defined. The relationship of SQL, JDBC and Java, specifically the underlying technologies and standards that Hibernate is built upon, are discussed. The software development challenges that embody the “object/relational paradigm mismatch” are individually identified. Additional generic problems that are encountered when creating object-oriented software clients to relational databases are also discussed. This module concludes with presenting the basic concepts of object/relational mapping and the capabilities provided by Hibernate.
2. **Getting started with Hibernate.** This module provides the student with the information necessary to complete a simple Hibernate project. The student will be exposed to the Hibernate core programming interfaces and will then be shown how to integrate a Hibernate application with “non-managed” and “managed” environments.
3. **Mapping Persistent Classes.** This module explains how to map the Plain Ordinary (Old) Java Objects of a rich domain model to Hibernate metadata. Mapping class inheritance hierarchies and fine-grained models will also be discussed. The module will conclude with an introduction to class association mappings.
4. **Working with Persistent Objects.** In this module, the student will learn about The lifecycle, or the various “states” of persistent objects in a Hibernate application. The student will begin working with the Session persistence manager and understanding transitive persistence.
5. **Transactions and Concurrency.** In this module, the student will learn how long-running application transactions can be associated with the Hibernate notion of a “unit of work”. Database transactions and locking will be discussed in the context of a long-running application transaction.
6. **Advanced Mapping Concepts.** The Hibernate type system and how it can be extended to create custom mapping types is one of the topics

explained in this module. The student is provided an introductory overview of advanced mapping concepts such as collection mapping and how to map one-to-one and many-to-many associations.

7. **Retrieving Objects.** In this module, the student is introduced to the basic concepts, concerns and means of retrieving object state from persistent store with Hibernate. This section introduces the student to fetching strategies and how to discern and solve the “n+1” Selects problem.
8. **Queries.** HQL, criteria, and native SQL queries are covered in detail. Once these foundational APIs are presented, the student is shown advanced reporting techniques, the use of dynamic queries and how to optimize runtime fetching
9. **Caching.** In this module, an explanation for the use of cache is provided. The student will learn about use of Hibernate first- and second-level cache and will be shown how the caching system is used in practice through examples from the CaveatEmptor application.
10. **Application Design and Architecture.** In this module, the student is introduced to some foundational best practices for writing layered applications with Hibernate. This section will address the use of Servlets and integration with Hibernate and EJBs in managed environments. Creating implementations using application transactions, handling legacy data and using audit logging are practices which are also addressed.
11. **The Hibernate Toolset.** This module is an introductory overview of the use of the Hibernate toolset and how it can be used in application development with Hibernate. The student will learn about automatic database schema generation and POJO code generation techniques. Importing legacy schemas with Middlegen and attribute-oriented programming with tools such as XDoclet will also be addressed.
12. **Hibernate, Java Persistence and EJB3.** This module is an introductory overview of the standardized Java Persistence API and its use with the EJB3 component framework. The student will be introduced to the Java Persistence API and will be provided a comparison of equivalent capabilities of the Hibernate API. The module then shows the use of the JPA in a managed environment using the new EJB3 POJO-based component framework. The student

will be shown how to write applications with the JPA via the new EJB3 Session and Entity Bean interfaces.