



## **Core java – Curriculum**

Advanced Java – J2EE Overview

### **Part 1:**

Java Database Connectivity (4.1)

Java Servlet (3.1)

Java Server Pages (2.3)

JSTL Tag Library

### **Part 2:**

Spring Framework (4.1.7)

Hibernate Framework (4.3.10)

Web Services → SOAP and REST Services (Apache CXF / Jersey)

## **Hibernate Training Course**

---

### **Introduction to Hibernate**

- JDBC VS Hibernate
- Overview of Hibernate Architecture
- Object Relational Mapping (ORM)
- [Creating Hibernate Application in Eclipse IDE](#)
- [Hibernate with annotation](#) and xml configuration example

Real Time Signals Technologies,  
#102, Krishna Grand, Over Marathahalli Bridge,  
Bangalore – 37, Phone: +91-8042008777, +91-9538566675, email: [info@realtimesig.com](mailto:info@realtimesig.com)



- Generator classes
- Hibernate Dialects

## **Object Lifecycle, Persistence, and Session Management**

- Hibernate Object Life cycle
- Transient State
- Persistent State
- Detached State
- Removed State
- Saving to Databases
- Persistent Context

## **Hibernate Session API**

- session.save(Object o)
- session.update(Object o)
- session.saveOrUpdate(Object o)
- session.merge(Object o)
- session.get(Object.class, Identifier)
- session.lock(Object, LockMode)
- session.delete(Object)
- session.clear()

## **Core interfaces of Hibernate**

- Session interfaces
- Session Factory interfaces
- Configuration interfaces
- Transaction interfaces
- Query and Criteria interfaces

## **Hibernate Mapping Associations using XML and annotations**

Real Time Signals Technologies,  
#102, Krishna Grand, Over Marathahalli Bridge,  
Bangalore – 37, Phone: +91-8042008777, +91-9538566675, email: [info@realtimesig.com](mailto:info@realtimesig.com)



- Lazy, Eager, Cascade, Inverse
- Fetching strategies
- One-to-one
- One-to-Many
- Many-to-one
- Bidirectional many-to-one
- Bidirectional one-to-one
- Many-to-many
- Bidirectional many-to-many

### **Collection (HAS-A) Mapping**

- Mapping List
- One-to-many Mapping using List
- Mapping Bag
- One-to-many Mapping using Bag
- Mapping Set
- One-to-many Mapping using Set
- Mapping map in Collection Mapping
- Many-to-many Mapping in hibernate using map

### **Mapping Inheritance(IS-A) using xml and annotations**

- Inheritance Mapping Strategies
- Table per Concrete Class
- Table per Subclass
- Table per Hierarchy

### **Hibernate Transaction Management**

- Hibernate transaction configuration
- Hibernate Transaction API
- Transaction Propagation



## **Enable Logging in Hibernate**

- [Hibernate Logging by log4j using log4j.xml file](#)
- [Hibernate Logging by log4j using log4j.properties file](#)

## **Hibernate Query Language**

## **Hibernate Criteria Query Language**

## **Hibernate Named Language**

## **Performance Tuning and Optimization**

- Hibernate Caching
- [Caching in Hibernate](#)
- [Hibernate Second Level Cache using EH Cache](#)
- Batching

## **Hibernate Connection Pool**

- Configure C3P0 connection pool in Hibernate
- Configure DBCP connection pool in Hibernate

## **Hibernate Integration with other Frameworks**

- [Hibernate and Spring Integration Tutorial](#)
- [Hibernate and Struts2 Integration Tutorial](#)

---

## **Hibernate Training Course**



---

## Introduction to Hibernate

- JDBC VS Hibernate
- Overview of Hibernate Architecture
- Object Relational Mapping (ORM)
- [Creating Hibernate Application in Eclipse IDE](#)
- [Hibernate with annotation](#) and xml configuration example
- [Generator classes](#)
- [Hibernate Dialects](#)

## Object Lifecycle, Persistence, and Session Management

- Hibernate Object Life cycle
- Transient State
- Persistent State
- Detached State
- Removed State
- Saving to Databases
- Persistent Context

## Hibernate Session API

- session.save(Object o)
- session.update(Object o)
- session.saveOrUpdate(Object o)
- session.merge(Object o)
- session.get(Object.class, Identifier)
- session.lock(Object, LockMode)
- session.delete(Object)
- session.clear()



## Core interfaces of Hibernate

- Session interfaces
- Session Factory interfaces
- Configuration interfaces
- Transaction interfaces
- Query and Criteria interfaces

## Hibernate Mapping Associations using XML and annotations

- Lazy, Eager, Cascade, Inverse
- Fetching strategies
- One-to-one
- One-to-Many
- Many-to-one
- Bidirectional many-to-one
- Bidirectional one-to-one
- Many-to-many
- Bidirectional many-to-many

## Collection (HAS-A) Mapping

- [Mapping List](#)
- [One-to-many Mapping using List](#)
- [Mapping Bag](#)
- [One-to-many Mapping using Bag](#)
- [Mapping Set](#)
- [One-to-many Mapping using Set](#)
- [Mapping map in Collection Mapping](#)
- [Many-to-many Mapping in hibernate using map](#)

## Mapping Inheritance(IS-A) using xml and annotations

- Inheritance Mapping Strategies
- Table per Concrete Class
- Table per Subclass
- Table per Hierarchy



## **Hibernate Transaction Management**

- Hibernate transaction configuration
- Hibernate Transaction API
- Transaction Propagation

## **Enable Logging in Hibernate**

- [Hibernate Logging by log4j using log4j.xml file](#)
- [Hibernate Logging by log4j using log4j.properties file](#)
- 

## **Hibernate Query Language**

## **Hibernate Criteria Query Language**

## **Hibernate Named Language**

## **Performance Tuning and Optimization**

- Hibernate Caching
- [Caching in Hibernate](#)
- [Hibernate Second Level Cache using EH Cache](#)
- Batching

## **Hibernate Connection Pool**

- Configure C3P0 connection pool in Hibernate
- Configure DBCP connection pool in Hibernate



## Hibernate Integration with other Frameworks

- [Hibernate and Spring Integration Tutorial](#)
- [Hibernate and Struts2 Integration Tutorial](#)

## Spring Framework

### Introduction to Spring Framework

- Benefits of Spring Framework
- Overview of Spring Framework Architecture
- Layer Architecture of Spring Framework

### IOC (Inversion of Control) Container

- Bean Factory **Container**
- Application Context Container
- Bean Definition
- Bean Scope
- Bean Life cycle
- Bean Post Processor
- Bean Inheritance
- 

### DI (Dependent Injection)

- Setter-based Injection
- Constructor-based Injection
- Injecting Inner bean
- Injecting collection
- Beans wiring
- Beans Auto-wiring

### Annotation based configuration

- @Required annotation
- @Autowired annotation



- @qualifier
- @PostConstruct and @PreDestroy
- @Resource
- @Configuration and @Bean

### **AOP (Aspect-Oriented Programming)**

- AOP Concepts
- SP AspectJ Annotation
- SP AspectJ XML

### **Transaction Management**

- **ACID** levels
- Local vs. Global Transactions
- Programmatic vs. Declarative
- **Transaction Methods**
- Isolation levels
- Propagation types
- Programmatic Transaction Management
- Declarative Transaction Management

### **Spring JDBC Framework**

- JDBC Template
- PreparedStatement
- ResultSetExtractor
- RowMapper
- NamedParameter
- SimpleJdbcTemplate
- SQL Stored Procedure in Spring
- Using DAO support
- Using AOP transaction support
- Applying Transaction support and integrating with DB

### **Spring Security**

### **Batch processing**



**E-Mail processing**

**Spring Integration with Hibernate**

**Spring Integration with Struts**

**Spring Web MVC Framework**

- Overview of Spring MVC Framework Architecture
- Handler mapping
- Controllers
- View Resolvers
- Annotations
- Log4j
- Validations
- Security
- JAXB
- Integration with Hibernate
- Exception Handling
- Restful Web Service