Java Persistence API (JPA) with EclipseLink
By Dr. Vishwanath Rao

Overview:
This intensive 5-day course provides participants with a comprehensive understanding of Java Persistence API (JPA) and its implementation with EclipseLink. Through a combination of lectures, hands-on exercises, and practical examples, participants will gain the knowledge and skills necessary to effectively utilize JPA for object-relational mapping in Java applications.

Intended Audience:
This class is ideal for Java developers, software engineers, and architects who want to enhance their understanding of JPA and learn how to leverage EclipseLink for ORM. It is also suitable for individuals seeking to deepen their knowledge of database integration and persistence in Java applications.

Prerequisites:
- Proficiency in Java programming language
- Familiarity with relational databases and SQL
- Basic understanding of object-oriented programming concepts

Duration:
5 days (40 hours)

Objectives:
By the end of this course, participants will:
- Understand the fundamentals of JPA and its role in Java applications.
- Learn how to configure EclipseLink as a JPA provider in Java projects.
- Gain proficiency in mapping Java objects to relational database tables using annotations.
- Master the concepts of entity relationships, inheritance mapping, and querying with JPQL.
- Explore advanced topics such as caching strategies, locking mechanisms, and performance tuning in EclipseLink.
- Acquire best practices for designing efficient JPA entities and repositories.
- Develop the skills to integrate EclipseLink with other Java EE technologies.
- Gain hands-on experience through practical exercises and

real-world examples.

What You Will Learn:

- Setting up EclipseLink in a Java project and configuring persistence.xml
- Mapping entity classes and relationships using JPA annotations
- Implementing inheritance mapping strategies with EclipseLink
- Executing JPQL queries for data retrieval and manipulation
- Utilizing the Criteria API for dynamic query construction
- Optimizing performance through caching and tuning techniques
- Managing transactions and handling concurrency with EclipseLink
- Testing JPA applications using JUnit and in-memory databases
- Implementing best practices for JPA entity design and repository management

Certification Details:

Participants who successfully complete the course and pass the final assessment will receive a certificate of completion. This certification validates their proficiency in JPA with EclipseLink and demonstrates their ability to design and implement robust persistence solutions in Java applications. Additionally, participants will have the opportunity to pursue further certifications in Java and database technologies to enhance their professional credentials.

Course contents

Day 1: Introduction to JPA and EclipseLink

- Overview of JPA and Object-Relational Mapping (ORM)
- Understanding the role of JPA in Java applications
- Introduction to EclipseLink as a JPA provider
- Setting up EclipseLink in a Java project
- Configuring persistence.xml for database connection
- Mapping Java objects to database tables using annotations

Day 2: Entity Relationships and Inheritance Mapping

- Defining entity relationships: @OneToOne, @OneToMany, @ManyToOne, @ManyToMany
- Bidirectional vs. unidirectional relationships
- Mapping inheritance hierarchies: Single Table, Joined, Table per Class
- Polymorphic queries and inheritance strategies in EclipseLink

Day 3: JPQL (Java Persistence Query Language) and Criteria API

- Introduction to JPQL and its syntax
- Executing JPQL queries using EntityManager
- Using JPQL for advanced querying, including joins and aggregates
- Introduction to Criteria API for type-safe querying
- Building dynamic queries with Criteria API in EclipseLink

Day 4: Advanced Topics in JPA and EclipseLink

- Caching strategies in EclipseLink: first level and second level cache
- Batch processing and performance tuning
- Optimistic and pessimistic locking
- Customizing EclipseLink behavior with EclipseLink extensions
- Integrating EclipseLink with other Java EE technologies like JTA and JMS

Day 5: Advanced Use Cases and Best Practices

- Transaction management and error handling
- Lazy loading vs. eager loading: optimizing fetching strategies
- Auditing and logging with EclipseLink
- Testing JPA applications with JUnit and in-memory databases
- Best practices for designing efficient JPA entities and repositories
- Case studies and real-world examples of JPA and EclipseLink usage