

# Oracle Coherence 12c: Share and Manage Data in Clusters Ed 1.1

**Course Duration: 32 Hours**

**Course code: Ed1.1**

## 1. Course Overview

This course provides a comprehensive understanding of Oracle Coherence 12c, focusing on building scalable, high-performance, and fault-tolerant clustered applications. Learners will gain hands-on experience in configuring distributed caches, managing data across clusters, and implementing real-time data grid solutions.

## 2. What you'll learn?

**By the end of the course, you will be able to:**

- Understand Oracle Coherence architecture and data grid concepts
- Install, configure, and manage Coherence clusters
- Implement distributed caching strategies
- Store, retrieve, and process data in-memory
- Develop and deploy Coherence-based applications
- Manage scalability, high availability, and fault tolerance
- Monitor and troubleshoot Coherence environments

## 3. Target Audience

**This course is ideal for:**

- Java Developers
- Middleware Administrators
- Application Architects
- System Integrators
- Cloud and Distributed Systems Professionals

## 4. Pre-Requisites

Before taking this course, you should have:

- Basic knowledge of Java programming
- Understanding of distributed systems concepts
- Familiarity with application servers (WebLogic preferred)
- Basic knowledge of networking concepts

## 5. Course content

Module 1: Course Introduction

- Course overview and objectives
- Introduction to in-memory data grids
- Oracle Coherence use cases

Module 2: Introduction to Oracle Coherence

- Overview of Oracle Coherence 12c
- Key features and benefits
- Coherence editions and licensing
- Real-world implementation scenarios

Module 3: Coherence Architecture

- Cluster architecture and components
- Storage-enabled and storage-disabled nodes
- Cluster communication and messaging
- Partitioning and data distribution

Module 4: Setting Up Coherence Environment

- Installation and configuration
- Cluster setup and initialization
- Configuration files (cache-config.xml, coherence.xml)
- Starting and managing nodes

## Module 5: Working with Distributed Caches

- Cache types and configurations
- Named caches and cache factories
- Data storage and retrieval
- Cache lifecycle management

## Module 6: Data Partitioning and Distribution

- Partitioned caches
- Backup and replication strategies
- Data affinity and locality
- Load balancing across cluster nodes

## Module 7: Entry Processing and Data Manipulation

- Entry processors
- Aggregators and filters
- Querying cached data
- Performing parallel processing

## Module 8: Events and Listeners

- Cache events and triggers
- Map listeners and interceptors
- Event-driven architectures
- Handling real-time updates

## Module 9: Persistence and Data Durability

- Coherence persistence options
- Snapshot and recovery mechanisms
- Disk-based storage integration
- Ensuring data durability

## Module 10: Transactions and Consistency

- Transaction management in Coherence
- Consistency models
- Concurrency control
- Managing data integrity

## Module 11: Integration with Applications

- Integrating Coherence with Java applications
- Using Coherence with WebLogic Server
- REST and API-based access
- Microservices integration

## Module 12: Scalability and High Availability

- Horizontal scaling strategies
- Failover and fault tolerance
- Cluster elasticity
- Performance optimization techniques

## Module 13: Monitoring and Management

- Monitoring tools and dashboards
- JMX and Coherence metrics
- Cluster health monitoring
- Logging and diagnostics

## Module 14: Security in Coherence

- Authentication and authorization
- Securing cluster communication
- Data encryption
- Role-based access control

## Module 15: Troubleshooting and Best Practices

- Common issues and resolutions
- Debugging cluster problems
- Performance tuning best practices
- Deployment guidelines

## Module 16: Final Project / Case Study

- Build a distributed caching solution
- Implement real-time data grid scenario
- End-to-end project evaluation