

## Course 472: Java Best Practices and Design Patterns (4 days)

This in-depth workshop provides developers and architects the ability to identify, apply and customize object-oriented (Gang-of-Four) and Java Enterprise design patterns. The course also introduces best practices in organizing and testing Java projects.

### Who can benefit?

This is an intermediate level Java programming course, designed for developers who wish to improve their Java programming and design skills. The student should be an experienced J2EE / Java programmer, with practical development experience in Java.

### Course Outline

#### Chapter 1: Improving Productivity

- Java coding conventions and best practices
- Exception best practices
- Equals and hashCode
- Automating building of Java applications
- How to create and run unit tests
- JUnit 3.8 and JUnit 4
- Unit testing from an IDE and Ant

#### Chapter 2: Designing Java Classes

- Design Principles
- Singleton
- Delegation vs. inheritance
- Template method
- Strategy
- Converting a method into an object
- Command
- Memento

#### Chapter 3: Creating behavior at runtime

- General-purpose classes
- Bridge
- Composite
- Visitor
- Role

#### Chapter 4: Creating objects on demand

- Limitations of constructors
- Simple Factory

- Prototype factory
- Inversion of control
- Builder
- Abstract Factory

### **Chapter 5: Improving code structure**

- API Design
- Adapter
- Façade
- Proxy
- Decorator
- Observer
- Chain of responsibility
- Mediator

### **Chapter 6: Optimization patterns**

- How to measure performance
- Profiling an application
- Monitoring an application
- Measuring response time
- Final keyword
- Weak references
- Arrays and collections
- Choosing right implementation of List, Set and Map
- Optimizing loops
- Loop order
- Flyweight
- Iterator vs. Random access
- Object pooling
- Double-checked locking and alternatives
- Read-write lock
- ThreadLocal

### **Chapter 7: Java EE design patterns**

- Tiered Architecture
- Model 2
- Intercepting Filter, Composite View, View Helper
- Session façade, Business Delegate
- Thread safety
- Data Access Object
- Service Oriented Architecture
- Proxy, Adapter
- Virtual Component

### **Chapter 8: Effective Java**

- Why effective Java?
- Creation and destruction
- Java.lang.Object
- Classes and interfaces

- Methods
- General programming tips
- Exception best practices
- Thread best practices

### **Chapter 9: Test-driven Development**

- What to test
- Forms of software testing
- Unit testing with JUnit
- Testing exception throwing
- Simplifying tests with JUnit4 annotations
- Integration and Functionality Testing
- Testing container-managed components
- Mock objects
- Testing Spring services
- Best practices to support mock objects
- Spring support for mock objects
- Testing Data Access Objects
- DBUnit

**Please contact your ROI representative to discuss course tailoring!**