

## Core Object-Oriented Design Patterns in Java (TT1200)

**Length:** 3 days

### Description

Throughout the course students will explore the following patterns, varying the levels of coverage to drill down on the most commonly used Patterns, and to simply survey others. Students will compare and contrast the patterns and explore the advantages and disadvantages of using certain patterns for explicit development functions in Java environments.

### Audience

This an introductory-level OO programming course, designed for developers who need to identify, design, and lead the implementation of OO projects. We will explore and apply the terminology, the specification, the processes and technologies specific to OO. Examples are written in Java.

### Prerequisite

Attendees should be familiar with UML and have basic programming experience in Java. This course is not recommended for developers new to Java programming.

### Topics

#### SESSION: INTRODUCTION TO DESIGN PATTERNS

- Overview of Patterns
- What is a Pattern?
- Why Patterns?
- Crucial Qualities of Good Patterns
- What are NOT Patterns?
- Classifying and Describing Patterns
- Overview of Patterns by Category

#### SESSION: CREATIONAL PATTERNS

- Class Creation vs. Object Creation
- Abstract Factory Pattern
- Description
- When to Apply
- Structure
- Example
- Benefits and Liabilities
- Issues
- Singleton Design Factory Pattern
- Standard Pattern Description
- Impact of Threading on Singletons

- Options for Addressing Threading Issues
- Challenge of Double-Checked Locking in Java
- Challenge of Clustered Servers
- Builder Design Pattern
- Standard Pattern Description
- Factory Method Design Pattern
- Standard Pattern Description
- Applying the Different Factory Patterns
- Prototype Design Pattern
- Standard Pattern Description
- Survey of Creational Patterns

#### SESSION: STRUCTURAL PATTERNS

- Overview of Structural Patterns
- Composite Pattern
- Standard Pattern Description
- Child Management
- Composite Pattern in Java Platform
- Adapter Pattern
- Standard Pattern Description

## Core Object-Oriented Design Patterns in Java (TT1200)

- Two-Way Adaptors
- Adaptor Pattern in Java Platform
- Proxy Pattern
- Standard Pattern Description
- Types of Proxies
- Proxy Pattern and Web Services
- Bridge Pattern
- Standard Pattern Description
- Façade Pattern
- Standard Pattern Description
- Decorator
- Standard Pattern Description
- Survey of Structural Patterns

### SESSION: APPLYING PATTERNS

- Selecting the Right Pattern
- Adapting an Existing Pattern
- Creating a New Pattern
- Implementation Considerations
- Challenges in Working With Patterns
- Anti-Patterns

### SESSION: BEHAVIORAL PATTERNS

- Observer Pattern
- Standard Pattern Description
- Java Built-in Support of Observer Pattern
- Observer Pattern in Java Platform
- Strategy Pattern
- Standard Pattern Description
- Iterator Pattern
- Standard Pattern Description
- Controlling Iteration
- Robust Iterators
- Aggregate Classes
- Java Iterator Implementation
- Visitor Pattern
- Standard Pattern Description
- Interpreter Pattern
- Standard Pattern Description
- Chain of Responsibility Pattern
- Standard Pattern Description
- Command Pattern
- Standard Pattern Description
- Command Pattern in Java Platform
- Mediator Pattern
- Standard Pattern Description
- State Pattern
- Standard Pattern Description
- Comparison and Summary