

Design Patterns in Java

Course Description:

This course presents examples, exercises, and challenges that enable you to grapple with alternative approaches and discover practical subtleties in applying design pattern theory to realistic problems. You learn to enhance your practical skills and build the confidence you need to use design patterns in mission-critical applications.

In this course, the patterns are organized into five major categories: interfaces, responsibility, construction, operations, and extensions. Each category begins with a chapter that reviews and challenges your ability to apply facilities built into Java. These introductory sections are followed by chapters that each explain a particular pattern, demonstrate the pattern in use with UML diagrams and Java code, and challenges for you to solve.

Course Objectives:

- Understand the purpose of Design Patterns
- Understand the various categories
- Create Java classes implementing some of the Design Patterns

Audience:

Java Programmers who need to create and maintain enterprise systems written in Java. This course is not intended to teach Java or enterprise applications such as JavaServer Pages (JSP), Servlets, Enterprise JavaBeans (EJB), HTML, XML, RMI and CORBA.

Prerequisites:

- Solid understanding of Object-Oriented concepts.
- Working knowledge of UML.
- Strong Java skills including the use of abstract classes, interfaces and packages.

Course Content:

Chapter 1 - About the Course

- Course contents
- Overview
- Audience
- Course objectives

Chapter 2 - Introduction to Patterns

- Defining design patterns
- Object oriented approaches

Chapter 3 - Creational Patterns

- The Factory Pattern
- How a factory works
- Example
- When to use a factory pattern

Chapter 4 - The Abstract Factory Pattern

- Example
- How the user interface works
- Consequences of abstract factory

Chapter 5 - The Singleton Pattern

- Example
- Static classes as singleton patterns
- Creating singleton using a static method
- Consequences of the singleton pattern

Chapter 6 - The Builder Pattern

- Example
- Consequences of the builder pattern

Chapter 7 - The Prototype Pattern

- Example
- Consequences of the prototype pattern

Chapter 8 - Structural patterns

- The Adapter Pattern
- Example
- Object adapter
- Class adapter
- Two way adapters
- Adapters in java

Chapter 9 - The Bridge Pattern

- Example
- Consequences of the bridge pattern

Chapter 10 - The Composite Pattern

- Example
- Consequences of the composite pattern

Chapter 11 - The Decorator Pattern

- Example
- Inheritance order
- Non-visual decorators
- Decorators, adapters and composites
- Consequences of the decorator pattern

Chapter 12 - The Façade Pattern

- Example
- Building the façade classes
- Consequences of the façade

Chapter 13- The Flyweight Pattern

- Example

Chapter 14 - The Proxy Pattern

- Example

Chapter 15 - Behavioral patterns

- Chain of Responsibility
- Applicability
- Example
- A chain or a tree?
- Consequences of the chain of responsibility

Chapter 16 - The Command Pattern

- Example
- Consequences of the command pattern
- Anonymous inner classes

Chapter 17 - The Interpreter Pattern

- Example
- Consequences of the interpreter pattern

Chapter 18 - The Iterator Pattern

- Enumerations in java
- Filtered iterators
- Example
- The filtered enumeration
- Consequence of the iterator pattern

Chapter 19 - The Mediator Pattern

- Example
- Initialization of the system
- Mediators and command objects
- Consequences of the mediator pattern

Chapter 20 - The Memento Pattern

- Implementation
- Example
- Consequences of the memento

Chapter 21 - The Observer Pattern

- Example
- Consequences of the observer pattern

Chapter 22 - The State Pattern

- Example
- Consequences of the state pattern

Chapter 23 - The Strategy Pattern

- Example
- Consequences of the strategy pattern

Chapter 24 - The Template Pattern

- Kinds of methods in a template class
- Example
- Templates and callbacks
- Consequences of the template pattern

Chapter 25 - The Visitor Pattern

- When to use the visitor pattern
- Example
- Double dispatching
- Traversing a series of classes
- Consequence of the visitor pattern