



Capstone Courseware, LLC

33 Boylston Street
Jamaica Plain, MA 02130

877-227-2477
capstonecourseware.com

192. Design Patterns in Java Software

Version 5.0

This course seeks to develop, for the experienced Java programmer, a strong, shared vocabulary of design patterns and best practices. The course begins with a discussion of how to recognize and apply design patterns - that is, how to incorporate pattern awareness into one's own analysis, design, and implementation practices. The main body of the course focuses on the Gang of Four design patterns, with a chapter each on creational, behavioral, and structural patterns. Classroom time is about evenly split between discussion, group design exercises, and coding labs to reinforce finer points of important patterns.

This is not a patterns catalog: it is as much a study of how to "think in patterns" as it is an introduction to several of the most important patterns. Students will be challenged to bring their own previous development experience to the discussion, to see the patterns in everyday design and coding solutions. The course puts more emphasis on some patterns than others. We believe that students will be better served by going into a few patterns in depth, with lively discussions of several others, than by following a regular routine of discussion and examples over every GoF pattern.

The course software also includes an optional overlay of workspace and project files to support use of the Eclipse IDE in the classroom. (This requires that the instructor be experienced in use of Eclipse and able to walk students through basic tasks in the IDE.)

This revision of the course targets the 5.0 version of the Java language and Core API. Students with Java 1.4 experience should find all of the concepts and most of the example and lab code accessible. (A few examples that involve the Collections API are necessarily heavy on generic types, and this may require some additional support from the instructor.)

Prerequisites

- Solid Java programming experience is essential - especially object-oriented use of the language. Language features and techniques that are integral to some lab exercises include interfaces and abstract classes, threading, generics and collections, and recursive methods. Course 103 is excellent preparation.



- Previous experience with UML (Unified Modeling Language) will be helpful, but is not critical. The course uses UML class diagrams extensively but keeps notation fairly simple, and also includes a quick-reference appendix.





Learning Objectives

- Start to think in terms of design patterns.
- Recognize and apply patterns to specific software development problems.
- Use known patterns as a shared vocabulary in designing and discussing solutions.
- Use Factories and Singletons to control object creation, for a variety of reasons.
- Use Observers, Observables, and Model/View/Controller systems to decouple application behavior and preserve code scalability.
- Understand the full motivation for the Command pattern and take advantage of Command frameworks in JFC.
- Implement Adapters, rather than building redundant classes or creating intermediate data structures for consumption by existing code.
- Understand and apply a range of other J2SE and J2EE patterns to improve code quality and scalability, and to produce high-quality solutions right off the bat.

Timeline: 3 days.

(A 2-day timeline for non-programmers is also possible.)

IDE Support: Eclipse 3.2

In addition to the primary lab files, an optional overlay is available that adds support for Eclipse 3.2. Students can code and build all exercises from within the IDE. Most exercises can be tested from within the IDE as well, though some must be tested from the command line. See also our orientation to Using Capstone's Eclipse Overlays, and please be advised that this is an optional feature; it is not a separate version of the course, and the course itself does not contain explicit Eclipse-specific lab instructions.





Chapter 1. Recognizing and Applying Patterns

- Design Patterns
- Defining a Pattern
- Unified Modeling Language
- Seeing Patterns
- Warning Signs and Pitfalls

Chapter 2. Creational Patterns

- Factory Patterns
- The Singleton Pattern
- APIs and Providers
- Cascading Factories

Chapter 3. Behavioral Patterns

- The Strategy Pattern
- The Template Method Pattern
- The Observer Pattern
- The Model/View/Controller Pattern
- The Command Pattern
- The Chain of Responsibility Pattern

Chapter 4. Structural Patterns

- The Composite Pattern
- The Adapter Pattern
- The Decorator Pattern
- The Façade Pattern
- The Flyweight Pattern

Chapter 5. J2EE Patterns

- Model/View/Controller, Redux
- The Intercepting Filter Pattern
- The Front and Application Controller Patterns
- The Business Delegate Pattern
- The Service Locator Pattern
- The Transfer Object Pattern
- The Composite Entity Pattern
- The Data Access Object Pattern





Appendix A. Learning Resources

Appendix B. UML Quick Reference

System Requirements

Hardware Requirements (Minimum)

500 MHz, 256 meg RAM, 500 meg disk space.

Hardware Requirements (Recommended)

1.5 GHz, 512 meg RAM, 1 gig disk space.

Operating System

Tested on Windows XP Professional. Course software should be viable on all systems which support a J2SE 1.4 SDK.

Network and Security

Limited privileges required -- please see our standard security requirements at <http://capcourse.com/Guides/Security.html>.

Software Requirements

All free downloadable tools.

