

Applying Advanced Object Designs with Patterns

Duration

3 days

Instructor

Craig Larman

Class Limit

20 students

Prerequisite

At least one year full-time hands-on object-oriented programming experience.

Price

On-site:

Please contact SPC for pricing (contact information on page 2)

Public Training:

\$1485 (3 days)

*Discount available for early registration

Materials Provided

- Student manual containing the course slides
- Student handouts with class exercises and class studies

This popular, high-impact, and hands-on course on mastering advanced object-oriented design (OOD) and design patterns is aimed at architects and developers of OO systems, and is based on acclaimed industry leader Craig Larman's extensive experience coaching and applying OOD for over 15 years.

In this intensive hands-on seminar there is some lecture time, but the majority is spent in high-value-education small teams "at the whiteboard" while Craig rotates and works with each team, guiding them through the case study problem. You will learn to design with patterns, apply visual modeling and the UML in an agile modeling approach, and a suite of related advanced design topics, including the design of packages.

The course opens with a single case study within which you learn design pattern skills. Other skills transfer methods include buzz groups, rotation reviews, multi-modal learning, and pair learning. These skills are applied again in a second, third, and fourth iteration of the same case study, as you go deeper, reinforce knowledge, and illustrate iterative development. New design patterns are introduced in each subsequent iteration.

Various studies indicate that after initial release of a software system, at least 50% of effort and cost is spent in modification. To save money, it is rational to take a long-term view of product development and invest in skillful designs that reduce these costs.

You will leave this seminar with deeper skill to apply patterns and create quality object-oriented designs that support reduced modification costs and increased comprehensibility. At their worst, patterns appear like purely disparate ideas, without much cohesion, or connection to your problems. At their best, they are an elegant and practical language of design, offering a vocabulary of reusable solutions at points of complexity, instability, variation, or evolution in your systems.

This seminar helps you acquire this cohesive vocabulary, and apply patterns at variation points. It provides examples and context for the most frequently used patterns, and includes a learning aid for guidance in their application. In short, you work hard and hands-on in this course. You will leave with the direct experience and confidence to actually do object design and apply patterns.

Essential, high-use UML and agile modeling is taught in the context of design modeling. However, the emphasis is on object design, not on learning diagramming syntax (UML notation). Each student receives a copy of *Design Patterns Explained*. And importantly, we actively work with the book continually throughout the course; it doesn't sit unused on a table. It is applied as a valuable and living learning aid that both supports the course directly, and will be appreciated as a long-term tool for reinforcing the course learning, and going further in object design.

After taking this course, you will be able to define and apply:

- the 23 "Gang of Four" (GoF) design patterns
- high-value "Pattern Languages of Program Design" (PLOP) patterns

TRAINING

Applying Advanced Object Design with Patterns

- high-value Enterprise Application Architecture patterns (from Martin Fowler)
- high-value J2EE and EJB design patterns, if the audience is using Java technologies
- design package structure for low dependency impact
- agile modeling and high-frequency UML

Intended Audience

Software architects and developers building object-oriented systems.



Instructor

Craig Larman is the author of *Applying UML and Patterns—An Introduction to OOA/D and the Unified Process*, the world's best-selling text on OOA/D, iterative development, and the UML, translated to many languages and used worldwide in industry and colleges. He also co-authored the *Java 2 Performance and Idiom Guide*, and is currently writing *Agile and Iterative Development: A Manager's Guide*.

He is known throughout the international software community as an expert and coach in object technologies, OOA/D, patterns, the UML, agile modeling, an agile approach to the Unified Process (UP), combining the UP with XP and Scrum practices, and iterative agile development methods.

He travels worldwide to fulfill his passion to serve people through coaching, speaking, and education, helping software organizations succeed with high-impact best practices such as iterative and agile methods, design patterns, automated continuous integration, and test-first development.

For more information on this or other SPC Springboard courses, please visit www.spcspringboard.com or e-mail SPC at info@spc.ca

Software Productivity Center
Suite 460—1122 Mainland Street
Vancouver, BC V8M 4T8
www.spc.ca

Toll Free:
1.877.548.1948

Fax:
604.689.0141

Vancouver:
604.662.8181

Toronto:
416.885.0512

Outline

The course is intensively exercise-based; you will leave with the confidence that comes from practice. Students collaborate in small groups at whiteboards working through a major case study problem (which has several iterations), while Craig visits the teams and assists. Between the practice sessions are short lectures, for just-in-time learning of topics related to the exercises. The lecture modules include:

- Design Patterns: Motivation and Categories
- Agile Modeling and Required UML
- Fundamental Architectural and Design Principles
- Patterns for Varying Single Algorithms
- Patterns for Varying Instance Behavior
- Patterns for Access Control
- Patterns for Structural Decomposition
- Patterns for Organization or Communication of Work
- Patterns for Varying Event Response
- Patterns for Varying Interfaces
- Patterns for Creation
- Patterns for Varying Implementations
- Patterns for Handling State
- Patterns for Varying Interactions
- Patterns for J2EE and EJB applications
- Patterns for Enterprise Application Architecture



TRAINING

