

Course: Design Patterns

Goals

Patterns reveal solutions to a problem that recurs in multiple contexts. Design patterns are proven solutions to software design problems that are independent of platform or language. With knowledge of design patterns, a designer or architect can leverage the collective knowledge and expertise of the software community. This course focuses on the 23 patterns in the classic Design Patterns book by the "Gang of Four" (GoF). Additional patterns include topic areas such as: Concurrency, Micro-Architecture, Pattern-Oriented Software Architecture (POSA), and Anti-Patterns.

At the end of the course, the student will be able to:

- Explain the role and value of patterns
- Articulate the Gang of Four patterns, and how they relate to each other.
- Discuss trade-offs in applying various design patterns, and when to use each pattern.
- Explain critically important design principles from Martin, Meyer and others.

Duration

Four days.

Prerequisites

Experience in software design or architecture is desirable, but not mandatory. At least 6 months of programming experience in Java, C# or C++ is highly desirable.

Cost

Please call **1-610-831-1151** for public enrollment and private, on-site pricing.

Description

This 4-day course is designed to provide students with a thorough introduction to, and understanding of, the most critical design patterns. From basic design principles the course builds a solid foundation for patterns for software design in diverse areas including application development, concurrency and software architecture. A special "anti-patterns" section describes patterns that we should not emulate.

Topics

Design Pattern Overview Critical Principles of Object-Oriented Design <ul style="list-style-type: none">– Meyer's Open-Closed Principle– Martin's Design Principles– SRP, ISP, DIP, etc. Martin's Principles of Package Architecture <ul style="list-style-type: none">– Package Cohesion– Package Coupling Basic Object-Oriented Design Patterns <ul style="list-style-type: none">– Delegation vs. Inheritance– Design to Interface– Null Object, etc. The 23 Gang of Four Patterns <ul style="list-style-type: none">– 5 Creational Patterns– 7 Structural Patterns– 11 Behavioral Patterns Micro-Architecture and System Patterns <ul style="list-style-type: none">– Object Pool– Worker Thread– Dynamic Linkage– Cache Management, etc.	Concurrency Patterns <ul style="list-style-type: none">– Single Threaded Execution– Guarded Suspension– Balking– Scheduler Pattern-Oriented Software Architecture <ul style="list-style-type: none">– Layers Architecture– Pipes & Filters Architecture– Blackboard Architecture– Broker– Model-View-Controller, etc. Process Patterns <ul style="list-style-type: none">– The Selfish Class– Patterns for Evolving Frameworks– Patterns for Designing in Teams– Patterns for System Testing Anti-Patterns (Learning from bad examples) <ul style="list-style-type: none">– Stovepipe System– Reinvent the Wheel– Golden Hammer– Death by Planning, etc.
------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Audience

Software developers, designers and architects who wish to learn advanced design techniques.

**For more information about this course or other courses please contact
Nazzaro & Associates at 1-610-831-1151.**