

## Design Pattern Developer Essentials

This engaging eight hour in-person or online training course, delivered in one or two sessions, presents a core set of design patterns every developer should be familiar with and explores the forces underlying patterns that make them valuable for building software.

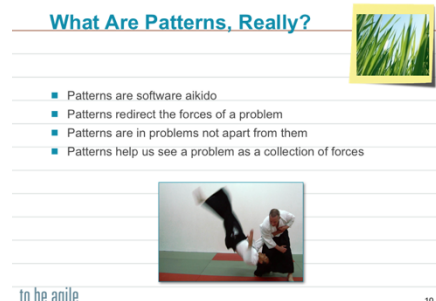
You'll see how understanding patterns reveals the essence of object-oriented thinking and creates a useful context for solving a variety of software problems. You'll come to understand patterns as more than just "reusable solutions to common problems within a given context" and recognize them as a collection of forces that help you penetrate deeper into problems, discovering elegantly simple solutions that make your software more robust and easier to maintain. The expanded view of patterns you'll gain from this course will help you encapsulate and abstract virtually any problem with maximum flexibility and without over-complicating the solution. You'll learn to make better coding choices and will master a shared vocabulary for talking about design that dramatically improves inter-team communication.

By the end of this training, you'll be armed with several new, effective tools for solving design problems that will empower you to produce immediate improvements in the quality of the software you design and build.

### Course Benefits

Completing this course will give you a deeper understanding of object-oriented development and enable you to:

- Explain what patterns are and know when to use them
- Read and write the 3 most-important UML diagrams
- Employ Agile principles to create higher-quality code
- Use a common vocabulary for communicating designs
- Adopt simple methods to find patterns in problems
- Understand patterns by what they encapsulate
- Apply patterns just-in-time to avoid over-design
- Appreciate the value of shared coding standards
- Contribute to design reviews and evaluate others' designs
- Gain several techniques for doing effective analysis and design
- Master techniques for emerging designs in iterative development



### Who Should Take This Course

This design patterns course is for all team members and has the greatest impact when the entire team attends. This course will benefit Architects, DBAs, Designers, Developers, Development Managers, Directors, Product Managers, Programmers, QA Engineers, Software Engineers, Technical Analysts, Technical Leads, Technical Writers, and Testers. Familiarity with basic Object-Oriented (OO) concepts and terminology is recommended.

## Agenda

<u>In One Full-Day (8-Hour) In-Person Session</u>	<u>In Two Half-Day (4-Hour) Online Sessions</u>
<p>Session 1 — Design</p> <p><b>Introduction:</b> Purpose, objectives, and logistics.</p> <p><b>Paradigms, Principles, and Perspectives:</b> Agile design, Gang of Four’s advice, design principles.</p> <p><b>Software Patterns:</b> Misconceptions, forces.</p> <p><b>Encapsulating Variable Behavior:</b> Strategy, Template-Method, State, and Bridge patterns.</p> <p><b>Encapsulating Foreignness:</b> Adapter and Façade.</p> <p><b>Design Exercise Case Study:</b> Presentation and discussion of a representative design problem and the use of patterns to solve it, examining the trade-offs of different patterns.</p> <p><b>Encapsulating Sequence and Cardinality:</b> Proxy, Decorator, Chain of Responsibility.</p> <p><b>Encapsulating Construction:</b> The importance of factories for object construction, and how patterns teach us to separate use from creation; examination of Factory-Method, Abstract Factory, and Singleton.</p>	<p>Session 1 — Design</p> <p><b>Introduction:</b> Purpose, objectives, and logistics.</p> <p><b>Paradigms, Principles, and Perspectives:</b> Agile design, Gang of Four’s advice, design principles.</p> <p><b>Software Patterns:</b> Misconceptions, forces.</p> <p><b>Encapsulating Variable Behavior:</b> Strategy, Template-Method, State, and Bridge patterns.</p> <p>Session 1A — Design</p> <p><b>Encapsulating Foreignness:</b> Adapter and Façade.</p> <p><b>Design Exercise Case Study:</b> Presentation and discussion of a representative design problem and the use of patterns to solve it, examining the trade-offs of different patterns.</p> <p><b>Encapsulating Sequence and Cardinality:</b> Proxy, Decorator, Chain of Responsibility.</p> <p><b>Encapsulating Construction:</b> The importance of factories for object construction, and how patterns teach us to separate use from creation; examination of Factory-Method, Abstract Factory, and Singleton.</p>

## Your Instructor, David Bernstein



My continuing passion for software design and construction has led me to train more than 10,000 developers in the last twenty-six years for clients that have included Fortune 500 firms such as Microsoft, IBM, Yahoo!, Boeing, AT&T, Sprint, Medtronic, SunGard, State Farm, MetLife and Weyerhaeuser. As a longtime IBM consultant, I trained software engineers around the globe, giving them the skills to write the next generation of applications and operating system

software while earning one of the highest satisfaction ratings in the history of IBM education. Since 2006, I've devoted my consulting practice to providing organizations with training and coaching for software developers and teams transitioning to Agile and Scrum.

## Certification

This course, part of our comprehensive Certified Scrum Developer Essentials Training, satisfies the one-day elective training requirement for becoming a Certified Scrum Developer (CSD) through the Scrum Alliance. This course counts for 8 Professional Development Units (PDUs). See <http://ToBeAgile.com/faq> for more details about this new certification.

## Praise for this Training

“I learned how to think effectively about code quality. I learned principles and practices of effective design. It explains patterns in plain English. David’s teaching style is very engaging and clear. I would highly recommend this class to any level of developer.”

—Todd Froyland, Senior Tech Lead/Architect

“This class is essential to any technical professional in a development environment. The approaches covered will help with every phase of the development cycle on any size team or project.”

—Tyler Ashbridge, Director of Systems Development

“We will be significantly faster at accommodating last-minute things that always seem to come up in our work. Even if you think it may not apply to you, do it. You will see how it does apply.”

—Martin Murphy, Senior Scientist