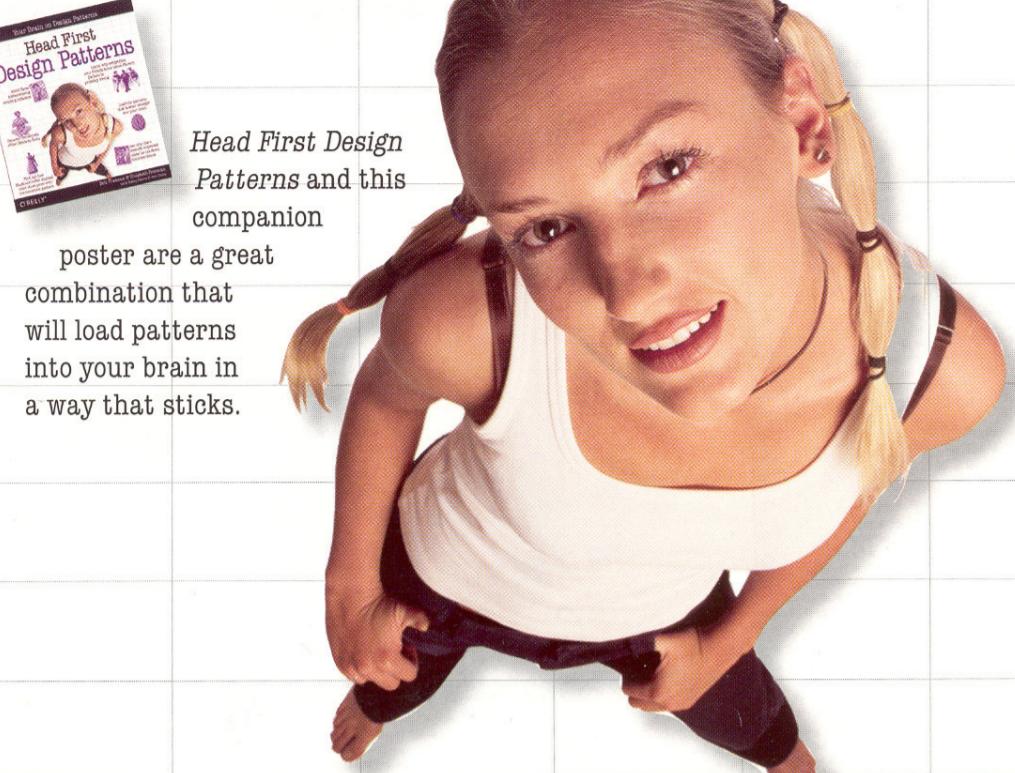


# Head First Design Patterns Poster



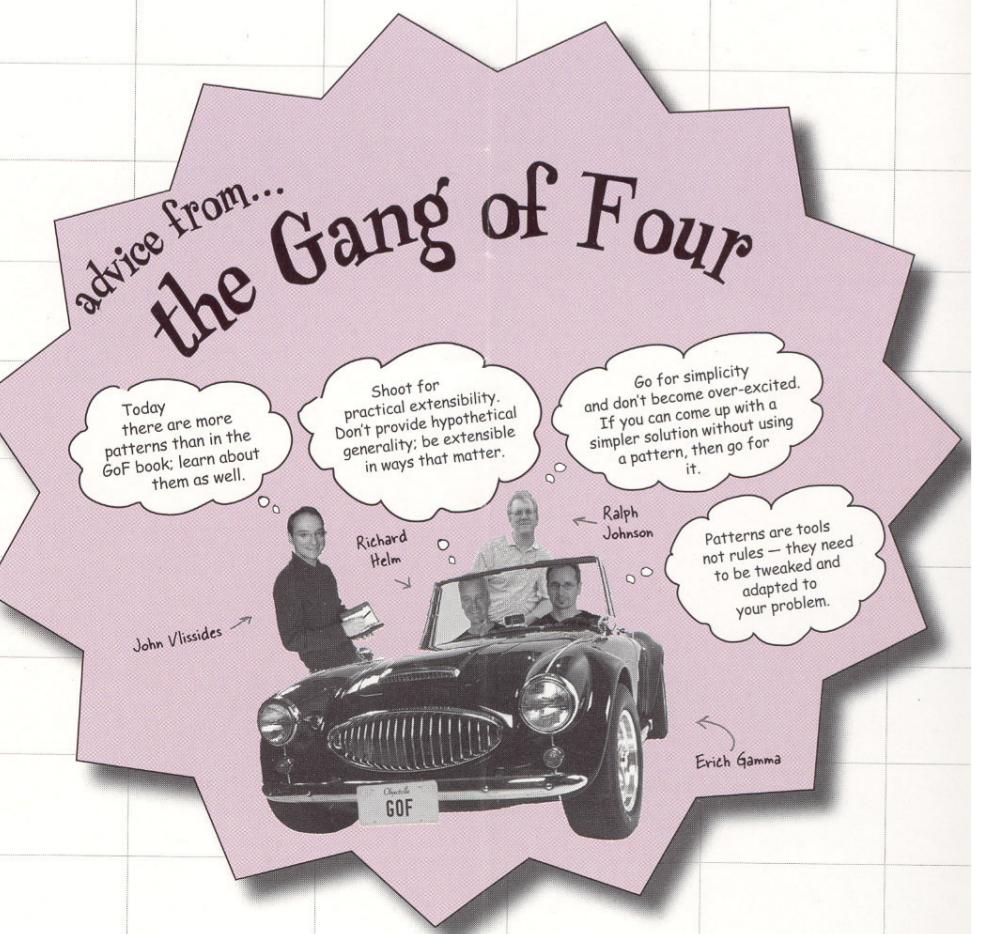
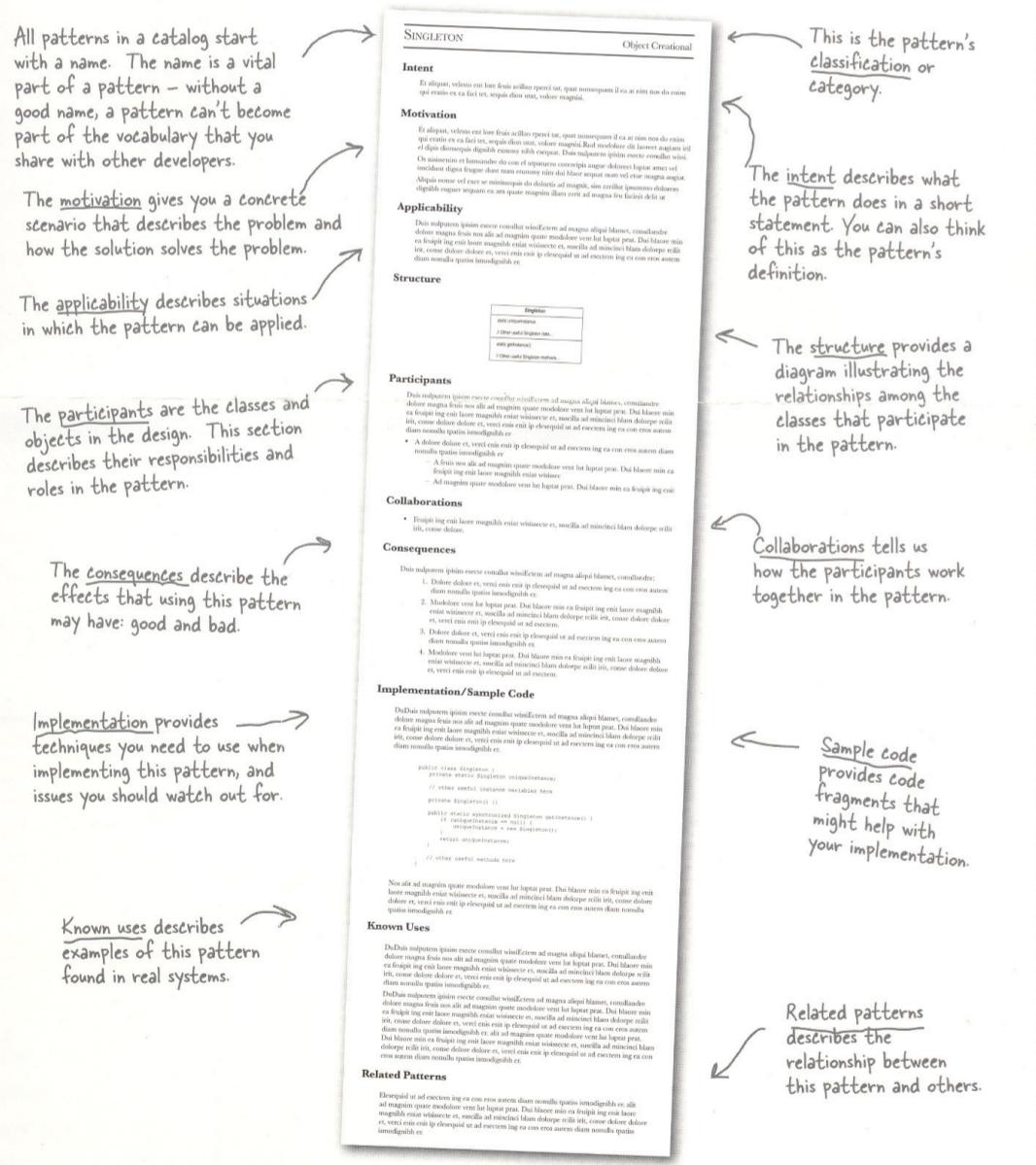
Head First Design Patterns and this companion poster are a great combination that will load patterns into your brain in a way that sticks.

Eric Freeman & Elisabeth Freeman

O'REILLY®

ISBN 0-596-10214-3 US \$9.95 CAN \$13.95  
9 780596 102142 6 36920 10214 4

## How to read a Patterns Catalog



**Factory Method** 134, 107

**Abstract Factory** 156, 87

**Adapter** 243, 139

**Command** 206, 233

**Decorator** 91, 175

**Façade** 264, 185

**Iterator** 336, 257

**Observer** 51, 293

**Proxy (Protection)** 474, 207

**Template Method** 289, 325

**MVC** 520, 4

**Simple Factory** 117, n/a

Page number in Head First Design Patterns  
Page number in the Gang of Four's Design Patterns

With Composite, we can create arbitrarily complex trees, and treat them as a whole...  
...or as parts!

Act now, and get the power of the shared vocabulary... yours for FREE!

Top five ways to share your vocabulary:  
1. In design meetings: When you're discussing design problems, make sure everyone is speaking the same language. This helps prevent misunderstandings and ensures that everyone is on the same page.  
2. With other developers: Share your vocabulary with your colleagues. This can help you communicate more effectively and avoid confusion.  
3. In architecture reviews: Use your vocabulary to explain the design decisions behind your system. This can help others understand the rationale behind the design.  
4. In code reviews: Use your vocabulary to review and critique code. This can help ensure that the code is consistent with the design and follows established conventions.  
5. In developer forums: Share your vocabulary in developer forums to help others understand the concepts you're discussing. This can help you find answers to your questions and contribute to the community.

**Composite** 356, 163

The Composite Pattern allows you to compose objects into tree structures to represent part-whole hierarchies. Composite lets clients treat individual objects and compositions of objects uniformly.

**Singleton** 177, 127

The Singleton Pattern ensures a class has only one instance and provides a global point of access to it.

**State** 410, 305

The State Pattern allows an object to alter its behavior when its internal state changes. The object will appear to change its class.

**Strategy** 24, 315

The Strategy Pattern defines a family of algorithms, encapsulates each one, and makes them interchangeable. Strategy lets the algorithm vary independently from clients that use it.

**Proxy (Virtual)** 462, 207

The Virtual Proxy acts as a representative for an object that may be expensive to create. The Virtual Proxy often defers the creation of the object until it is needed, and also acts as a surrogate for the object before and while it is being created.

**MVC** 520, 4

A Compound Pattern, the model-view-controller pattern (MVC) separates an application into three distinct components: the model, view, and controller.

**Simple Factory** 117, n/a

Not a true Pattern, but like Abstract Factory and Factory Method, commonly used to encapsulate object creation.

**Anti-Pattern**

Name: Golden Hammer

Problem: You need to choose technologies for your development and you believe that exactly one technology must dominate the architecture.

Context: You need to develop some new system or piece of software that doesn't fit well with the technology that the development team is familiar with.

Forces:

- The development team is committed to the technology they know.
- The development team is not familiar with other technologies.
- Unfamiliar technologies are seen as risky.
- It is easy to plan and estimate for development using familiar technology.

Suggested Solution: Use a familiar technology anyway. The technology is applied obsessively to many problems, including places where it is clearly inappropriate.

Refactored Solution: Develop a knowledge of study groups that expose developers to new solutions.

Example: Web companies keep using and maintaining their internal homegrown caching systems when open source alternatives are available.

Adapted from the Portland Pattern Repository's WIKI at <http://c2.com> where you'll find many anti-patterns and discussions.

**What's Inside...**

The poster in your hand is a companion to Head First Design Patterns book. It summarizes visually 18 of the most common design patterns including Adapter, Command, Composite, Decorator, Observer, Strategy, and more. Each pattern includes a handy page reference to both Head First Design Patterns and the "Gang of Four" text, the canonical description of the pattern, and a visual guide designed (and tested by the examples in Head First Design Patterns) to jog your memory of the objects, classes and their relationships.

**Your Brain on Design Patterns**

Head First Design Patterns

O'REILLY®