

Chapter 1: Design Concepts

You will learn:

- Improving C# applications.
- Reverse-engineering designs from C#.
- Choosing the right pattern.
- Improving design quality incrementally.
- Making design tradeoffs.

Chapter 2: Applying Patterns to the .NET Environment

You will learn:

- Networks.
- Databases.
- Interactive applications.
- Integrating frameworks.
- Creating designs systematically with UML.
- Improving architectures with patterns.

Chapter 3: Architectural Principles

You will learn:

- Good top-level architecture.
- Avoiding architectural pitfalls.
- Deploying tiered architecture.
- Deploying layered architectures.
- Architectural tradeoffs.

Chapter 4: Applying Design Patterns

You will learn:

- Choosing the right design pattern.
- Recognizing design patterns in C# applications.
- Delegation versus inheritance.
- Adapting standard design patterns to applications.
- Creating interfaces.

Chapter 5: External Interfaces

You will learn:

- How to simplify package interfaces with facade.
- Mediating composite object interaction.
- Bridging interface incompatibilities.

Chapter 6: Process and Network Connections

You will learn:

- Process connectivity.
- RPCs.
- Messages.
- RMI.
- Pipe and socket subsystems.

Chapter 7: Designing Internal Interfaces

You will learn:

- Providing a surrogate using proxy.
- Creating object trees.
- Analyzing interface tradeoffs.
- Constructing event-driven applications.

Chapter 8: Designing Views

You will learn:

- Defining and organizing views.
- Supporting multiple views with observer.
- Applying MVC: Model-View-Controller.
- Architectures.
- Adapting the .NET framework.



Chapter 9: Choosing Controllers

You will learn:

- Combining views and controllers.
- Procedural controller.
- Event-driven controller.
- Minimizing controller complexity.
- Decoupling senders from receivers.

Chapter 10: Chain of Responsibility Pattern

You will learn:

- Encapsulating requests as objects.
- Command patterns.
- Event handling in .NET framework.

Chapter 11: Saving and Restoring State

You will learn:

- Generating objects using factories.
- Cloning objects using prototype.
- Managing class hierarchies with builders.
- Sharing state with flyweight.

Chapter 12: Connecting to Databases

You will learn:

- Decoupling databases from applications.
- Accessing objects in collections.
- Simplifying system control.

Chapter 13: Control Strategies

You will learn:

- Control tables and control structures.
- Interpreting commands for extensibility.
- Consolidating behavior with state machines.



Chapter 14: Managing Behavior

You will learn:

- Implementing roles in C#.
- How to use the strategy pattern.
- Changing object classes using state.
- Adding properties and operations dynamically.

Chapter 15: Reverse- and Re-engineering OO Systems

You will learn:

- Managing changing requirements.
- Designing for performance.
- Reverse-engineering legacy applications.



TM