

KOENIG SOLUTIONS · COURSE OUTLINE

Software Architecture: Meta and SOLID Principles in C#

Learn how to develop maintainable software systems
applying Design Patterns based on Meta and SOLID Principles

2 Days

DURATION

6

MODULES

52

TOPICS

COURSE OVERVIEW

This course examines the SOLID design principles in C#, beginning with the Single Responsibility Principle and moving through the Open/Closed, Liskov Substitution, Interface Segregation and Dependency Inversion Principles, in each case identifying common violations and refactoring towards a better design. Dependency Inversion is covered in depth, including dependency injection techniques, pure DI, IoC containers and their architectural implications. The course closes by examining metaprinciples such as DRY, KISS, YAGNI, Separation of Concerns and Command Query Separation, and how they relate to and sometimes conflict with SOLID.

WHO THIS COURSE IS FOR

Software Developers

Software Architects

Application Designers

Technical Leads

PREREQUISITES

Prerequisites

- The candidate must have knowledge of C#.
- The candidate must have knowledge on .net framework and visual studio 2019.

LEARNING OBJECTIVES

- Explain the Single Responsibility Principle and identify its violations
- Apply the Open/Closed Principle to reduce common design smells
- Analyse Liskov Substitution Principle violations and related contracts
- Refactor designs that violate the Interface Segregation Principle
- Apply Dependency Inversion using dependency injection techniques
- Compare pure DI with IoC-container approaches to building applications
- Describe metaprinciples such as DRY, KISS and YAGNI
- Compare SOLID principles against YAGNI and each other

6 Modules, 52 Topics

01

Single Responsibility Principle

TOPICS

SOLID Intro · SRP Definition. Problem Statement · Demo of the Problem · Refactoring to a Better Design · More Examples of SRP Violations · SRP Related Patterns

02

Open/Closed Principle

TOPICS

Outline · OCP Definition. Problem Statement · Demo of the Problem · Refactoring to a Better Design · OCP Related Patterns · Common Smells of OCP Violation

03

Liskov Substitution Principle

TOPICS

Outline · LSP Definition. Problem Statement · Contracts · Demo of the Problem · Refactoring to a Better Design · More Examples of LSP Violations · Common Smells of LSP Violation

04

Interface Segregation Principle

TOPICS

Outline · ISP Definition. Problem Statement · Demo of the Problem · Refactoring to a Better Design · Demo of the Problem. Example 2 · Refactoring to a Better Design. Example 2 · Common Smells, Fixes and Related Patterns

05

Dependency Inversion Principle

TOPICS

- Outline
- DIP Definition. Problem Statement
- Dependencies
- Volatile and Stable Dependencies
- IoC and DI Definitions
- DIP Violation Demo
- Refactoring to a Better Design Applying Dependency Injection (DI)
- DI Techniques
- Architectural Implications
- Pure DI and IoC-Containers
- Building a Simple IoC-Container
- Demo of a Real-World App Built with an IoC-Container
- Common Smells of DIP Violations

Metaprinciples and SOLID

TOPICS

- Outline
- DRY - Don't Repeat Yourself
- KISS - Keep it Simple, Stupid
- YAGNI - You Ain't Gonna Need It
- SoC - Separation of Concerns
- CQS - Command Query Separation Principle
- Principle of Least Astonishment
- Encapsulation and Information Hiding
- General Principles of Building APIs
- SOLID VS YAGNI
- OCP VS YAGNI
- SRP and ISP. What's the Difference?
- Architecture and Design

WHAT IS INCLUDED

Everything You Need to Succeed

- Koenig course material for every module
- Delivery by certified instructors and subject matter experts
- Hands-on exercises throughout the course
- Course completion certificate from Koenig

ABOUT KOENIG

Who We Are

Established in 1993, Koenig Solutions is a globally recognised training provider with 35+ technology partnerships, 5,000+ training programmes and a network of certified instructors and subject matter experts, supporting individuals and enterprise teams worldwide.

FLEXIBLE DELIVERY

Learn the Way That Suits You

[Classroom Training](#)[Live Online Classes \(ILO\)](#)[Fly-Me-A-Trainer \(FMAT\)](#)[Flexi \(Self-Paced\)](#)[1-on-1 Training](#)

Ready to Get Started?

Talk to a learning advisor about schedules, group training and customised delivery.

[Visit koenig-solutions.com](https://www.koenig-solutions.com)[✉ info@koenig-solutions.com](mailto:info@koenig-solutions.com)