C# SD - C# e SOLID: Principi e Design Patterns per lo Sviluppo Software Moderno

ITCore Sviluppo - C# - Sviluppo

Durata:	Lingue:	Certificazione:
3 Giorni	**Italiano**	**-**

Descrizione del corso

Il corso **"C# e SOLID"** è pensato per sviluppatori provenienti da qualsiasi linguaggio di programmazione, che desiderano approfondire i **principi di progettazione SOLID** e i **design patterns** più utilizzati nel settore. Il programma fornisce competenze pratiche per creare applicazioni modulari, manutenibili e scalabili, applicando metodologie universalmente riconosciute per una buona architettura del software.

A Chi è Rivolto Questo Corso

Questo corso avanzato è dedicato a sviluppatori con una buona esperienza in C# e conoscenza dei concetti base della programmazione orientata agli oggetti. È ideale per chi desidera portare la propria competenza a un livello successivo, padroneggiando le tecniche di design pattern per risolvere problemi architetturali e costruire software di alta qualità.

Obiettivi del Corso

Al termine del corso, i partecipanti saranno in grado di:

• Applicare i **principi SOLID** per migliorare la progettazione del software.
• Comprendere e implementare **design patterns** per risolvere problemi comuni in modo elegante e riutilizzabile.
• Creare architetture software scalabili, flessibili e aderenti alle best practices moderne.
• Trasferire e applicare i concetti trattati ad altri linguaggi di programmazione oltre a C#.

Destinatari:

• Sviluppatori software con conoscenza di base di programmazione orientata agli oggetti.
• Professionisti provenienti da altri linguaggi (Java, Python, C++) interessati ad applicare design patterns e principi architetturali in **C#**.

Un percorso completo e pratico per padroneggiare **C#**, implementare soluzioni software eleganti e prepararsi alle sfide architetturali più complesse.

Programma

- **Principi SOLID**
  - **SRP**: Single Responsibility Principle
  - **OCP**: Open/Closed Principle
  - **LSP**: Liskov Substitution Principle
  - **ISP**: Interface Segregation Principle
  - **DIP**: Dependency Inversion Principle
- **Design Patterns Creazionali**
  - **BU**: Builder
  - **FA**: Factory
  - **PR**: Prototype
  - **SI**: Singleton
- **Design Patterns Strutturali**
  - **BR**: Bridge
  - **CO**: Composite
  - **DE**: Decorator
  - **FA**: Façade
  - **FL**: Flyweight
  - **PR**: Proxy
- **Design Patterns Comportamentali**
  - **CH**: Chain of Responsibility
  - **CO**: Command
  - **IN**: Interpreter
  - **IT**: Iterator
  - **ME**: Mediator
  - **MM**: Memento
  - **ON**: Observer
  - **ST**: State
  - **TE**: Template
  - **VI**: Visitor
  - **ST**: Strategy
- **Pattern di Architettura e Ottimizzazione**
  - **R**: Repository
  - **UW**: Unit of Work
  - **LA**: Lazy Load