.Net Best Practices and Design Patterns

Category: Software Development & Programming | **Vendor:** Various

Duration: 32.00 hours (4 days) **26.0 CPD Hours** **Rating:** ⭐ 4.6 (5,878 reviews)

Course Information

Delivery Format: Instructor Led - Online

Course Overview

Gain the skills needed to solve real-world software development problems, and deliver fast, reliable applications. In this training course, you will identify and solve common design and architecture problems, apply best practices and design patterns to .NET application development, write code that is more maintainable, flexible and resilient, and apply design patterns for application reuse and flexibility.

About This Course

Gain the skills needed to solve real-world software development problems, and deliver fast, reliable applications. In this training course, you will identify and solve common design and architecture problems, apply best practices and design patterns to .NET application development, write code that is more maintainable, flexible and resilient, and apply design patterns for application reuse and flexibility.

Who Should Attend

This course uses Visual Studio 2013, but is useful to those using an earlier version

Prerequisites & Entry Requirements

General Prerequisites:

- Programming experience at the level of: .Net Development Introduction to C# Programming (CPROG), C# Programming, or Course Visual Basic Programming for .Net (VBP) Visual Basic Programming for .NET

- You should have experience: ◦Working with Visual Studio

- Using object-oriented concepts

- Building .NET applications in Visual Basic, C#, or C++

Learning Outcomes

Upon successful completion of this course, participants will be able to:

- Solve complex programming problems using design patterns

- Deliver bug-free code using test-driven development in Visual Studio

- Create layered architectures for reusability and to eliminate functional duplication

- Apply the S.O.L.I.D. principles for better class design

Detailed Course Outline

Simplifying Complex Programming with Proven Design Patterns

Applying simple interfaces to intricate algorithms

- Unifying interfaces from subsystems for ease of use and reusability

- Implementing the Facade pattern

Varying functionality by programming to interfaces

- Creating abstractions to enhance application adaptability and flexibility

- Exploiting the Strategy pattern

Extending object behaviour dynamically

- Increasing functionality without impacting existing code

- Composing objects with the Decorator pattern

Achieving reuse and flexibility

- Eliminating code duplication by outlining an algorithm

- Employing the Template Method pattern

Interfacing incompatible classes

- Transforming an interface to add value to existing code

- Harnessing the Adapter pattern

Applying Test–Driven Development Techniques

Automating unit testing

- Shortening development cycles with automated tests

- Eliminating regression errors with reusable tests

Integrating testing and coding

- Applying the test–first programming practice

- Generating immediate red–green feedback for increased code quality and shortened feature development cycles

- Organising, coordinating and running test cases

- Isolating a class test environment with the Mock–Object pattern for reliable and repeatable testing

- Tracking the Red–Green–Refactor rhythm for feedback

Refactoring code to improve design

- Improving design with behaviour preserving refactorings

- Eliminating code duplication by refactoring to patterns

Architecting a Layered Application

Designing the application architecture

- Layering architectures for reusability, durability and scalability

- Decoupling object creation with the Factory pattern

- Preserving object identity with the Identity Map pattern

Programming application tiers

- Isolating UI layers with the MVC and MVVM patterns

- Organising state–rich applications with the State pattern

- Restructuring database tables without impacting code

Modelling a business area

- Liberating rich Business Object Models from database structures using the Domain Model pattern

- Mapping rich Business Objects to database tables with the Data Mapper pattern

- Collapsing Business Object hierarchies with the Inheritance Mapper pattern

- Exploiting declarative programming in Microsoft Entity Framework to implement Data Mapping classes

Applying Best Practices

Organising and implementing business logic

- Managing Persistence Ignorant Entity types with Entity Framework

- Manipulating groups of Entities with business logic classes

Applying S.O.L.I.D. principles of class design

- Single Responsibility

- Open–Closed

- Liskov Substitution

- Interface Segregation

- Dependency Inversion

Automating Repetitive Tasks

Easing data access code

- Reducing database access code by exploiting Entity Navigation Properties

- Eliminating database update code with Entity change tracking

Enhancing application code

- Automating design reviews with FxCop and Code analyser

- Reducing User Interface code with Data Binding

Skills You Will Learn:

- Solve complex programming problems using design patterns

- Deliver bug-free code using test-driven development in Visual Studio

- Create layered architectures for reusability and to eliminate functional duplication

- Apply the S.O.L.I.D. principles for better class design

Frequently Asked Questions

Q: What delivery options are available for .Net Best Practices and Design Patterns?

We offer multiple delivery formats:

- Live Instructor-Led Classroom Online (Virtual/Live Online)
- Traditional Instructor-Led Classroom Training (ILT)
- On-site delivery at your offices anywhere in United Kingdom
- Private dedicated courses customized for your team

Q: How many CPD hours does this course provide?

The 4-day .Net Best Practices and Design Patterns course provides up to 26.0 CPD hours of structured learning. CPD certificates can be provided upon request.

Q: What is the duration of the .Net Best Practices and Design Patterns training?

The training takes place over 4 day(s), with each day lasting approximately 32.00 hours including breaks for lunch and refreshments.

Q: Do you provide corporate training for .Net Best Practices and Design Patterns?

Yes, we provide corporate training, dedicated training, and closed classes for .Net Best Practices and Design Patterns. Training can take place anywhere in United Kingdom including London, Manchester, Birmingham, Edinburgh, or live online allowing teams from across United Kingdom or internationally to attend.

Q: What related terms do people search for?

Popular related searches include: .Net Best Practices and Design Patterns, .NET Development, BPDP

Q: Why choose Nexus Human for .Net Best Practices and Design Patterns?

Nexus Human is recognized as one of the leading training providers. Our trainers have won multiple awards including:

- Small Firms Best Trainer Award
- National Training Partner of the Year (Ireland) - Multiple Years
- Global Top 30 Instructor Awards (2012, 2019, 2021)
- Tech Excellence Award Nominations
- Learning Performance Institute (LPI) External Training Provider Sponsor 2024

Q: Are there any discount codes available?

Yes! Use discount code **PENPAL5** when booking your .Net Best Practices and Design Patterns training. Please note that only one discount code can be used per booking and cannot be combined with other special offers.

Nexus Human

Professional Training & Development

Email: info@nexushuman.com

Website: www.nexushuman.com

Phone: +353 1 889 8700 (Ireland) | +44 20 8142 8961 (UK)

© 2026 Nexus Human. All rights reserved. This brochure was generated on 13/11/2026.