

## .NET Framework using C# (OI4112)

**Duration:** 4 days

### Description

This four-day course is designed to provide a sound introduction to the .NET Framework for programmers who already know the C# language and the fundamentals of Windows Forms. It is current to .NET 3.5 and Visual Studio 2008. The course focuses on core portions of the .NET Framework that are common across many application areas.

The course starts with an introduction to the architecture and key concepts of .NET. It then discusses class libraries, assemblies, versioning, configuration, and deployment, which constitute a major advance in the simplicity and robustness of deploying Windows applications, ending the notorious "DLL hell." The next two chapters discuss important topics in the .NET programming model, including metadata, reflection, I/O, and serialization. The following chapter continues the discussion of the .NET programming model, covering memory management, threading, asynchronous programming, application domains, marshal by value, marshal by reference, and .NET remoting.

.NET Security is introduced in some detail, including both code access security and role-based security. The next chapter covers interoperability of .NET with COM and with Win32 applications. The course includes an introduction to database programming using ADO.NET and LINQ. SQL Server 2005 Express Edition, bundled with Visual Studio 2008, is used in database examples. Finally, the .NET Framework diagnostic facilities are discussed in depth.

### Audience

Programmers who know the C# language and the fundamentals of Windows Forms will benefit from this course.

### Prerequisites:

The student should be an experienced application developer or architect with a working knowledge of C#, including building simple GUIs with Windows Forms.

### Topics

#### .NET FUNDAMENTALS

- What is Microsoft .NET?
- Common Language Runtime
- Attribute-Based Programming
- Interface-Based Programming
- Metadata
- Common Type System
- Framework Class Library
- Language Interoperability

- Managed Code
- Assemblies and Deployment
- Web Services
- ASP.NET

#### CLASS LIBRARIES

- Components in .NET
- Building Class Libraries at the Command Line

## **.NET Framework using C# (OI4112)**

- Class Libraries Using Visual Studio 2008
- Using References

### **ASSEMBLIES, DEPLOYMENT AND CONFIGURATION**

- Assemblies
- Private Assembly Deployment
- Shared Assembly Deployment
- Configuration Overview
- Configuration Files
- Programmatic Access to Configuration
- Using SDK Tools for Signing and Deployment
- Application Settings with .NET 2.0

### **METADATA AND REFLECTION**

- Metadata
- Reflection
- Late Binding

### **I/O AND SERIALIZATION**

- Directories
- Files
- Serialization
- Attributes

### **.NET PROGRAMMING MODEL**

- Memory Management and Garbage Collection
- Threading and Synchronization
- Asynchronous Delegates
- .NET 2.0 BackgroundWorker
- Application Domains
- Marshal by Value
- Marshal by Reference
- .NET Remoting

### **.NET SECURITY**

- Authentication and Authorization
- Configuring Security
- Code Access Security

- Code Groups
- Evidence
- Permissions
- Role-Based Security
- Principals and Identities

### **INTEROPERATING WITH COM AND WIN32**

- .NET Client Calling a COM Server
- PInvoke

### **ADO.NET AND LINQ**

- ADO.NET Overview
- .NET Data Providers
- Connections
- Commands
- DataReaders and Connected Access
- Data Sets and Disconnected Access
- Language Integrated Query

### **DEBUGGING FUNDAMENTALS**

- Compile-time Errors and Run-time Errors
- Configuring Debug, Release, and Special Builds
- Visual Studio 2008 Debugger
- Just-In-Time Debugging

### **TRACING**

- Tracing
- Event Logs

### **MORE ABOUT TRACING**

- Using the BooleanSwitch and TraceSwitch Classes
- Print Debugging Information with the Debug Class
- Instrumenting Release Builds with the Trace Class
- Using Listeners
- Implementing Custom Listeners