Programming the .NET Framework 4.0/4.5 with C# 5.0

Course 70240 – 40 Hours

Overview

Microsoft's .NET Framework presents developers with unprecedented opportunities. From 'geo-scalable' web applications to desktop and mobile platform applications - all can be built with equal ease, using substantially the same skill-set. But to make the most of this potential, developers must have a thorough grasp of core language skills. This course is intended for developers who will use C# to write .NET Framework applications and who are new to the C# programming language and the .NET Framework .
This includes those currently working with Visual Basic 6.0, C++ or Java. This 5-day intensive workshop concentrates on the C# programming language, The Common Language Runtime (CLR) and the .NET base class library, to prepare delegates fully in readiness for exploring the .NET Framework. From basic procedural syntax to sophisticated object-oriented programming techniques, delegates will learn how to write .NET applications with code that is robust, maintainable and efficient. The course is presented as a mixture of lectures and hands-on exercises. Practical sessions follow each topic, designed to reinforce the points covered .

On Completion, Delegates will be able to

- Write efficient procedural code that includes sequence, selection and iteration constructs
- Create and use classes and structures (types), including fields, properties and methods
- Use private, internal, protected and public scope
- Create derived classes that inherit from custom-written or .NET Framework classes
- Create interfaces and apply techniques of polymorphism effectively and appropriately
- Build exception-handling into methods, to create robust, user-friendly applications
- Understand custom attributes and their uses
- Use advanced C# features
- Understand and use generic constructs
- Understand garbage collection mechanics in the CLR
- Use solid object oriented techniques for writing solid C# code

Who Should Attend

The course is intended for anyone migrating to the .NET framework, and wants to gain a solid, robust understanding of the technology and its application with the C# language

Prerequisites

- Delegates must understand the fundamentals of programming and should have had prior Experience in programming
- Experience in object oriented programming is highly beneficial

JOHN BRYCE
Leading in IT Education
a matrix company

matrix

Course Contents

Module 1: Introduction to .NET and C#

- What is .NET?
- The Common Language Runtime (CLR)
- The Common Type System (CTS)
- Introduction to C#
- Namespace and Assemblies basics
- Viewing metadata with ILDasm and Reflector
- Introduction to Visual Studio
- Creating a simple C# Console Application
- .NET Overview – from .NET 1.0 to .NET 4.5

Module 2: C# Language Fundamentals

- Procedures and Statements
- Data Types
- Declaring Variables
- The var keyword
- Assignments
- Conversions
- Arithmetic and Other Operators
- Control Constructs

Module 3: Types

- Type Concepts
- Value Types vs. Reference Types
- Fields, Properties and Methods
- Method Overloading
- Default and Optional Arguments
- Accessibility Modifiers
- Automatic Properties
- Construction and Assignment
- The Simple Types
- The null Reference
- Static and Instance Members
- Enumerated Types
- Partial Classes
- Static Classes
- Nested Types

Module 4: Inheritance

- What is Inheritance?
- Extending a Class
- Polymorphism
- Upcasts and Downcasts
- Virtual and Override Modifiers
- New and Sealed Modifiers

Module 5: Abstract Classes and Interfaces

- Abstract Classes
- Abstract Methods and Properties
- Interfaces
- Interfaces and Polymorphism
- Standard Interfaces: IEnumerable, IComparable, IComparer
- Side Casts
- The is and as Operators
- Multiple Interfaces
- Explicit Interface Implementation

Module 6: Arrays, Collections and Strings

- Arrays
- Initializing Arrays
- The Array Class
- Multi-dimensional Arrays
- Jagged Arrays
- Indexers
- Standard Collections: ArrayList, Stack, Queue, Hashtable
- The String Type
- String Members
- The StringBuilder Type
- String Literals

Module 7: Exceptions

- Errors vs. Exceptions
- Error Handling Options
- The try block
- The catch block
- The throw statement
- The finally block
- Standard Exception Classes
- Custom Exceptions
- Checked and Unchecked Expressions
- Exception Handling Guidelines

Module 8: Generics

- The Need for Generics
- Generic Types
- Standard Generic Collections
- Generic Methods
- Generic Interfaces
- Generic Constraints
- Nullable Types
- Other Aspects of Generics

Module 9: Reflection and Attributes

- Metadata and Reflection
- Getting Information about Types
- The Type Class
- Dynamic Invocation
- Dynamic Creation
- Custom Attributes
- Applying Attributes
- Setting and Querying Attributes
- Introduction to the Managed Extensibility Framework (MEF)

Module 10: Delegates and Events

- Delegate Basics
- Creating Delegates
- Invoking Delegates
- The Delegate and MultiCastDelegate Types
- Anonymous Delegates
- Generic Delegates
- Events
- The Publisher / Subscriber Pattern

Module 11: Managing Resources

- Garbage Collection and its Impacts
- The Managed Heap
- Object Creation and Destruction
- The Garbage Collection Process
- Finalization
- Deterministic Finalization
- The IDisposable Interface
- The Dispose Pattern
- The using keyword
- The GC Class
- GC Types
- Other issues with Resource Management

Module 12: Namespace and Assemblies

- Namespaces
- The using keyword (with namespaces)
- Assemblies
- Assembly loading
- The Global Assembly Cache (GAC)
- Deploying Assemblies
- Versioning and Probing

Module 13: Advanced Language Constructs

- Partial Methods
- Iterators
- Extension Methods
- Lambda Expressions
- Object and Collection Initializers
- Anonymous Types
- Introduction to LINQ

Module 14: C# 4.0

- Default and Named Arguments
- Dynamic Binding and dynamic keyword
- Custom Binding
- Generic Co- and Contra-Variance

Module 15: Data Streams and Files

- The Stream Abstract Class
- The FileStream Class
- File I/O
- The File and FileInfo Classes
- The Path Class
- The Directory and DirectoryInfo Classes
- Stream Readers and Writers

Module 16: Debugging and Tracing

- The Debug Class
- The Trace Class
- Debug vs. Release Builds
- The AnyCPU Configuration
- Advanced Tracing
- The DebugView Tool

Module 17: Threading

- Processes and Threads
- Threading Basics
- The Thread Class
- Starting Threads
- Synchronization Basics
- The lock keyword
- Other Threading Issues
- The Asynchronous Programming Model (APM)
- Asynchronous Delegates

Module 18: C# 5.0

- Asynchronous Programming with C# 5.0
- The Await keyword