Advanced .NET Framework Programming Using C#

Duration: 35 hours

Prerequisites: C# programming experience.

Course Description: This hands-on course examines how to utilize advanced features of C# and the .NET Framework in order to build sophisticated, scalable, high-performing applications. The course includes coverage of features available in .NET 2.0 through .NET 4.8.

The course begins by reviewing .NET's Common Type System, and then examines nullable types, inferred types and dynamic data. Advanced object-oriented programming topics include auto-implemented properties, inheritance, abstract classes, sealed classes, and generics, as well as how to implement many of the .NET interfaces in order to take advantage of .NET functionality.

Students learn how to use synchronous and asynchronous delegates to call methods using late binding, as well as how to use delegates to define and fire custom events and manage callbacks. The course shows how to build multithreaded applications and synchronize access to shared resources, including the Thread and ThreadPool classes as well as .NET 4.0's Task class and the Parallel Task Library. Coverage of the .NET 4.5 Async and Await features is included.

Students learn how to work with data in .NET's collections, as well as how to create and use custom generic methods and collections. Database topics include how to use ADO.NET to manipulate data in databases and how advanced ADO.NET features provide support for transaction management, connection pooling, and the management of disconnected DataSets. Coverage includes how to work with XML documents and make XPath queries.

The course includes the use of LINQ to make queries of data, regardless of its location. Coverage includes using LINQ to Objects, LINQ to SQL, LINQ to DataSets and LINQ to XML, as well as how PLINQ can be used to make efficient queries on large sets of data located in memory.

Students learn about .NET's support for n-tiered application development, including encapsulating functionality in private and shared assemblies. Students also explore how to create and consume WCF services to build distributed systems.

The course includes coverage of attributes and reflection, and how to leverage attributes and reflection to design creative, robust solutions to common design problems.

Other topics include: overriding System.Object methods; boxing and unboxing issues; when to use the String vs. StringBuilder types; working with data in other cultures (internationalization); and pattern matching using regular expressions.

Comprehensive labs provide students with extensive experience coding with Visual Studio to practice with the topics presented throughout the course.

Students Will Learn

- .NET Type Fundamentals
- Object-Oriented Programming
- Using LINQ
- Threading - Building and Synchronizing Threads

- String Manipulation
- Working with Collections
- Delegates and Events
- ADO.NET - Connected Access
- ADO.NET - Disconnected Access
- Processing XML
- Threading - Working with the Thread Pool
- Threading - Parallel Programming
- Building Assemblies
- Attributes and Reflection
- Windows Communication Foundation Overview

Overview

Working with Types

- Common Type System
- Value vs. Reference Types
- Using Nullable Types
- Using Inferred Types
- Working with Dynamic Data

Working with Text

- Using Strings and `StringBuilder`
- Measuring Performance Using Stopwatch
- Working with Data from Other Cultures
- Using Pattern Matching and Regular Expressions

Working with Delegates

- Understanding Delegates
- Single-cast vs. Multi-cast Delegates
- Defining and Using Delegates
- Working with Synchronous and Asynchronous Delegates
- Understanding the Relationship Between Delegates and Events
- Defining and Raising Custom Events
- Handling Custom Events

Working with XML Data

- Understanding XML and XML Schemas
- Using XML with DataSets
- Using `XmlReader` and `XmlWriter`
- Manipulating XML Data Using `XmlDocument`
- Querying XML Data Using XPath
- Working with `XPathDocument`

Object-Oriented Programming

- Defining and Using Classes
- Understanding Partial Classes
- Using Inheritance
- Abstract Classes vs Sealed Classes
- Understanding `System.Object` Methods
- Understanding .NET Interfaces
- Working with Interfaces

Collections and Generics

- Understanding .NET Collections
- Ordered vs. Unordered Collections
- Managing Data with .NET Collections
- Defining Generic Methods
- Building and Using Generic Classes
- Defining Extension Methods

Managing Data with ADO.NET

- ADO.NET Object Model
- Connected vs. Disconnected Access
- Working with Data in Databases
- Calling Stored Procedures
- Working with Transactions
- Managing Connection Pooling
- Using the ADO.NET Provider Factory
- Working with Untyped DataSets
- Working with Typed DataSets
- Managing DataViews

Working with LINQ

- Understanding LINQ
- Building LINQ Queries
- LINQ and Extension Methods
- Defining Data Layers Using LINQ
- Using LINQ to Objects
- Using LINQ to SQL
- Using LINQ to DataSets

- Using LINQ to XML

Working with Threads

- Overview of Threading
- Creating Threads
- Passing Data to Threads
- Returning Data from Threads
- Managing Threads
- Problems with Threads
- Synchronizing Threads
- Debugging Threads
- Using Async and Await

Working with the ThreadPool

- Understanding Thread Pooling
- Managing the ThreadPool
- Creating Threads Using the `ThreadPool` Class
- Working with the Task Class
- Managing Tasks

Working with the Parallel Task Library

- Overview of Parallel Programming
- Using the `Parallel` Class
- Making PLINQ Queries
- Using Concurrent Collections

Assemblies

- Understanding .NET Assemblies
- Private vs. Shared Assemblies
- Where is the Global Assembly Cache (GAC)?
- Building and Using Private Assemblies
- Defining Strong Names
- Building, Installing and Using Shared Assemblies
- Configuring Assemblies
- Targeting Multiple Versions of .NET

Working with Attributes

- Understanding Attributes
- Using .NET Attributes
- Designing Custom Attributes
- Using Custom Attributes

Using Reflection

- Understanding Reflection
- Extracting Type Information
- Using Reflection for Late Binding

Windows Communication Foundation (WCF)

- Understanding WCF
- Defining Service and Data Contracts
- Building a Service
- Hosting a Service
- Managing Endpoints
- Exposing Metadata
- Calling a WCF Service

Related Bootcamp

Track	Duration	Price
Microsoft .NET Developer: C#	161 hours	$4,475

Contact Us

Address: 6 Hemlock Drive Chelmsford, MA 01824

Phone: 978.250.4983

Mon - Thur: 9 am - 5 pm EST
Fri: 9 am - 4 pm EST

E-mail: info@developer-bootcamp.com

Copyright© 2025 Developer Bootcamp