WCF Programming

Students Will Learn:

- .NET Framework Architecture
- C# Language Syntax
- SOA Benefits
- WCF Architecture
- Building Service Contracts
- Working with Data Contracts

- WCF Security Options
- Hosting WCF Services
- Choosing WCF Bindings
- Managing Service Instances
- Fault Handling
- WCF Routing Control

Course Description: This course provides students with hands on experience using Visual Studio to create service-oriented applications using Windows Communication Foundation (WCF) and C#. This class provides a thorough introduction to the C# programming language, including coverage of the essentials of the C# programming language, built in data types, operators, control structures, classes and methods. Students then learn how to leverage the power of the .NET Framework to build Web Service applications that interoperate with consumer applications including other platforms and technologies.

Students will learn how to configure addresses, bindings, and service and data contracts as well as how to use various techniques for developing endpoints to allow communication between consumer applications and the web services provider.

The course includes coverage of instance management, fault handling, and security. Students will learn how to use the WCF Routing Service for load balancing, content-based routing, and protocol bridging.

Comprehensive labs and exercises provide the students with experience creating both content server and consumer applications.

This course provides thorough coverage of the use of **Windows Communication Foundation** for service-oriented applications. Students requiring additional coverage of **ASP.NET Web Forms**, **Windows Forms** or **Windows Presentation Foundation** should contact SPIRO or refer to SPIRO complete course listing for additional training courses.

server and consumer applications.

This course provides thorough coverage of the use of **Windows Communication Foundation** for service-oriented applications. Students requiring additional coverage of **ASP.NET Web Forms**, **Windows Forms** or **Windows Presentation Foundation** should contact SPIRO or refer to SPIRO complete course listing for additional training courses.

Course Prerequisites: Prior experience with a scripting or programming language is required.

WCF Programming Using C# Course Overview:

Introduction to .NET

- Overview of the .NET Framework
- How .NET is Different from Traditional Programming
- Common Language Runtime (CLR)
- Common Language Specification (CLS)
- Common Type System (CTS)
- .NET Assemblies
- Microsoft Intermediate Language (CIL)
- .NET Namespaces
- .NET Framework Class Library

Introduction to Visual Studio

- Setting Profiles
- Creating a Project
- Using the Code Editor
- Setting Project Properties
- Adding References
- Compiling a Program
- Running a Program
- Debugging a Program
- Using the MSDN (Help)

Language Fundamentals

- C# Program Structure
- Defining Namespaces
- Understanding C# Data Types
- Defining Variables and Constants
- Comparing Value Types versus Reference Types
- Working with Operators and Expressions
- Performing Type Conversions
- Using Console I/O
- Formatting Numbers, Date and Times

Conditionals and Looping

- if/else
- switch
- while and do/while
- for
- foreach

Functions and Parameters

- Defining Static and Instance Functions
- Passing Parameters by value and by reference
- Overloading Functions
- Optional Parameters
- Using Variable Length Parameter Lists

Collections

- Defining and Using Arrays
- Understanding System.Array
- Using .NET Collections
- Working with ArrayLists and Hashtables
- Working with Lists and Dictionaries
- Introducing LINQ

Introduction to WCF

- WCF Web Services Architecture
- Addresses, Bindings and Contracts
- WCF Service Libraries
- WCF Test Host and Test Client
- ChannelFactory Class
- Configuring WCF Clients
- Standard Endpoints

Exception Handling

- What are Exceptions?
- .NET Exception Hierarchy
- Catching Exceptions
- Throwing Exceptions
- Managing Resources with Finally

Object-Oriented Programming

- Overview of Object-Oriented Programming
- Defining and using Classes
- Extending .NET Classes via Inheritance
- Defining and Implementing Interfaces
- Understanding the Role of Interfaces in .NET
- Working With Enumerations

Service Addresses

- Address Types
  - Endpoint Address
  - Base Address
  - MEX (Message Exchange) Address
- Metadata Exchange
- Address Formats

Selecting Binding Options

- Binding Selection
- HTTP Bindings
- TCP and Named Pipe Bindings
- MSMQ Binding
- BasicHttpBinding Class

Defining Service Contracts

- Service and Operation Contracts
- Creating Contracts at the Class and Interface Level
- Using ServiceContractAttribute
- Types of Service Contracts
  - Oneway
  - Request-Reply
  - Duplex
- Callbacks
- Asynchronous Proxies
- WSDL Files
- Contract Inheritance and Overloading
- Implementing Message Exchange Patterns
- Versioning

Endpoints

- Endpoints Explained
- Working with Endpoints
- Configuring Endpoints
- Using Multiple Endpoints

Managing a Service Instance

- Configuring Behaviors
- Service Instance Models
  - Per-Call
  - Per-Session
  - Singleton
- Threading Considerations
- Consuming WCF Application Services with .NET Applications
- Consuming WCF Application Services on foreign platforms

Defining Data Contracts

- Using DataContractAttribute
- Mapping Data to Schema
- Returning Arrays
- Returning Generic Collections
- Data Serialization
- Versioning

Fault Handling

- FaultException class
- FaultCode class
- FaultContract class
- Client Exception Handling
- Including Exception Details

Securing WCF Applications

- Security Issues with Services
- Types of Security
  - Transfer Security
  - Transport Security
  - Message Security
- Configuring Security on Client and Server
- Managing Certificates
- Configuring Client Certificates
- Sending Credentials

WCF Routing Configuration

- WCF Routing Service
- Hosting the Service
- Consuming the Service
- Service Contract and Implementation
- Routing Contracts
- Message Filters
- Common Routing Scenarios
  - Load Balancing
  - Content Based Routing
  - Service Partitioning
  - Protocol Bridging