C# Developer

Duration : 5 days

Course Overview

The C# Developer course comprises sessions dealing with C# applications, variables and operators, decision and loop constructs, arrays, enums, methods, objects, classes, structs, delegates, events, object-oriented principles, exception handling, file IO, LINQ.

Exercises and examples are used throughout the course to give practical hands-on experience with the techniques covered.

Target Audience

This course is aimed at anyone who wants to learn C# as a first language and developers who want to migrate to C# from another language, particularly those with little or no object-oriented knowledge.

Prerequisites

Delegates attending this course should have some previous programming experience and be able to define general programming concepts including compilation, execution, variables, arrays, sequence, selection and iteration, navigate the filesystem (on the command line ideally), edit and save text files and browse the web. This knowledge can be obtained by attendance on the pre-requisite Introduction to Programming course.

Objectives

This course aims to provide the delegate with the knowledge to be able to produce simple C# applications that exploit the core elements of the language including variables, expressions, selection and iteration, arrays, methods, classes and objects, structs, encapsulation, inheritance and polymorphism, exceptions, file IO and LINQ.

The delegate will learn and acquire skills as follows:

- Writing, compiling, and executing simple C# applications
- Understanding the relationship between C# and .NET
- Declaring and initialising variables
- Constructing expressions using relational, arithmetic and logical operators
- Constructing decision and loop statements
- Constructing and manipulating arrays and enums
- Declaring and invoking methods using both parameters and return values
- Creating classes, objects and structs
- Encapsulating data
- Implementing a hierarchical application design using inheritance
- Exploiting polymorphism
- Creating generic collections
- Using delegates and events
- Constructing simple lambda expressions
- Throwing and catching exceptions
- Reading and writing files
- Using Language Integrated Query (LINQ)

Course Content

Session 1: Introducing C#
History of the Language
C# and .NET
Common Intermediate Language
Common Language Runtime
Compiling and Running Code
Development tools

Session 2: Variables
Using Namespaces
Declaring Variables
Value and Reference Types
Datatypes
Variable Scope

Session 3: Operators
Arithmetic Operators
Unary Operators
Concatenation Operators
Comparison Operators
Logical Operators

Session 4: Decisions
If Statements
Switch Statements

Session 5: Loops
Types of Loop
Break and Continue Statements

Session 6: Arrays and Enums
Populating Arrays
Looping Through Arrays
Working with Enums

Session 7: Methods
Fields and Methods
Method Parameters
Method Overloading
Optional Parameters

Session 8: Objects
Creating Objects
Instance Fields
Instance Methods

Session 9: Classes and Structs
Class Structure
Constructors
Using Structs

Session 10: Encapsulation
Access Modifiers
Getters and Setters
Properties

Session 11: Static Members
Static Fields
Static Methods
Static Classes
Static Context

Session 12: Inheritance and Polymorphism
Object Oriented Programming
Extending Classes
Inheritance Modifiers
Polymorphism

Session 13: Abstraction and Interfaces
Abstract Methods
Abstract Classes
Using Interfaces
Interface Polymorphism

Session 14: Exception Handling
Throwing Exceptions
Catching Exceptions
Using try, catch and finally
Custom Exceptions

Session 15: Strings and Dates
The String Class
Using StringBuilder
Date Handling
Formatting Numbers

Session 16: Collections
Working with Collections
Generic Collections
The List Class
The Dictionary Class

Session 17: Delegates, Lambdas and Events
Creating Delegates
Lambda Expressions
Events and Event Handlers

Session 18: File IO
I/O Streams
Handling Files
Serialising and Deserialising Data

Session 19: LINQ
Query Expressions
Ordering and Projection
Grouping and Joins
LINQ to XML