Learnsoft
ENTERPRISE TRAINING

Learning to Program with C#

Length: 3 Days

Prerequisites: This course assumes that students have some programming background. No specific experience with Visual Studio 2008 or the .NET Framework is required. As with any such course, the more experience you bring to the course, the more you'll get out of it. This course moves quickly through a broad range of programming topics, but it does not require any prior .NET skills.

Summary: Students will learn the fundamental elements of programming classes, objects and methods. They will learn structured programming techniques, arrays and data structures, exception handling, string handling, fundamental algorithms and GUI programming concepts.

COURSE CONTENT

FUNDAMENTALS OF THE PROGRAM DEVELOPMENT CYCLE

- Computer Architecture
- The Notion of Algorithms
- Source Code vs. Machine Code
- Compile-Time vs. Run-Time
- Software Program Architecture
    - Standalone
    - Client/Server
    - Distributed
    - Web-Enabled
- IDE (Interactive Development Environment) Concepts

APPLICATION DEVELOPMENT FUNDAMENTALS

- Structure of a C# Program
- Memory Concepts
- Fundamental Data Type Declarations
- Fundamental I/O Concepts
- Fundamental Operators
- Arithmetic Operators
- Logical Operators
- Precedence and Associatively
- Building and Deploying a C# Program

INTRODUCTION TO CLASSES AND OBJECTS

- Classes, Objects and Methods
- Object Instances
- Declaring and Instantiating a C# Object
- Declaring Methods
- set and get Methods
- Initiating Objects with Constructors
- Value Types vs. Reference Types

FLOW CONTROL

- Conditional Constructs
- Looping Constructs
- Counter-Controlled Repetition
- Sentinel-Controlled Repetition
- Nested Control Constructs
- break and continue Statements
- Structured Programming Best Practices

WRITING METHODS (FUNCTIONS)

- Static vs. Dynamic Allocation
- Declaring Methods
- Declaring Methods with Multiple Parameters
- Method-Call Stack

- Scope of Declarations
- Argument Promotion and Casting
- Designing Methods for Reusability
- Method Overloading

ARRAYS

- Purpose of Arrays
- Declaring and Instantiating Arrays
- Passing Arrays to Methods
- Multidimensional Arrays
- Variable-Length Argument Lists
- Using Command-Line Arguments
- Using Environment Variables

DEEPER INTO CLASSES AND OBJECTS

- Controlling Access to Class Members
- Referencing the Current Object Using this
- Overloading Constructors
- Default and No-Argument Constructors
- Composition of Classes
- Garbage Collection and Destructors
- The finalize Method
- Static Class Members

DEFINING CLASSES USING INHERITANCE

- Superclasses and Subclasses
- Advantages of Using Inheritance
- protected Class Members
- Constructors in Subclasses

INCREASING CONVENIENCE BY USING POLYMORPHISM

- Purpose of Polymorphic Behavior
- The Concept of a Signature
- Abstract Classes and Methods
- Non-Inheritable Methods and Classes
- Purpose of Interfaces
- Using and Creating Interfaces
- Common Interfaces of the .NET Framework

FILES AND STREAMS

- Concept of a Stream
- Class File
- Sequential Access
- Object Serialization to/from Sequential Access Files
- Using LINQ to Query Information from a File

EXCEPTION HANDLING

- Types of Exceptions
- Exception Handling Overview
- Exception Class Hierarchy
- Extending Exception Classes
- When to Throw or Assert Exceptions

FORMATTED OUTPUT

- Conversion Characters
- Specifying Field Width and Precision
- Controlling Appearance with the IformatProvider Interface
- Printing Literals and Escape Sequences
- Formatting Output with the String.Format Method

STRINGS, CHARACTERS AND REGULAR EXPRESSIONS

- Fundamentals of Characters and Strings
- String Class
- String Operations
- StringBuilder Class
- Char Structure
- Regular Expressions
- Regular Expression Syntax
- RegExp Class
- Match Class
- System.Text.RegularExpressions Namespace

FUNDAMENTAL GUI PROGRAMMING CONCEPTS

- Overview of Windows Forms
- Displaying Text and Graphics in a Window
- Implementing Event Handlers
- Control Properties and Layout
- Mouse Event Handling
- Implementing Menus