

C# .NET COURSE CONTENT

Introduction to C# and NET

- ❖ Object Oriented Programming
- ❖ C#: The OOP Language
- ❖ The NET Framework
- ❖ CLR and Managed Code
- ❖ MSIL and JIT
- ❖ Metadata
- ❖ Assemblies
- ❖ Garbage Collection
- ❖ Putting Things Together
- ❖ ILASM and ILDASM
- ❖ A First Look at the C# Code
- ❖ The C# Code
- ❖ The IL Code
- ❖ The Manifest
- ❖ Using the Library File
- ❖ How to Get a Free C# Compiler
- ❖ Compiling Programs in the Command Line Environment
- ❖ If You Have the Compiler without the IDE
- ❖ If You Have the Visual Studio IDE
- ❖ Comparison of C# and C++
- ❖ The Features of C#
- ❖ The New Features of C#

Building the Program Logic

- ❖ Using Conditions
- ❖ Relational Operators
- ❖ Logical Operators
- ❖ The Logical AND Operators (&&,&)
- ❖ The Logical OR Operators (||,|)
- ❖ The Logical NOT Operator (!)
- ❖ The Bitwise Operators
- ❖ The ifelse Construct
- ❖ Manipulating Characters
- ❖ Nested ifelse Statements
- ❖ The switch Construct
- ❖ The Conditional Expression
- ❖ Using Libraries
- ❖ Repetition Loops
- ❖ The for Loop
- ❖ Using continue and break
- ❖ Available Options in the for Loop
- ❖ Nesting Loops
- ❖ The while Loop
- ❖ The do while Loop
- ❖ Branching Statements
- ❖ Arrays
- ❖ One Dimensional Arrays
- ❖ Declaring and Initializing Arrays
- ❖ Multi Dimensional Arrays
- ❖ Jagged Arrays
- ❖ Accessing Array Elements
- ❖ Using Program Arguments
- ❖ Using NET Properties and Methods with Arrays
- ❖ Array's Length (Length)
- ❖ Array's Rank (Rank)
- ❖ Sorting an Array (Array Sort)
- ❖ Reversing an Array (Array Reverse)
- ❖ Resizing an Array (Array Resize)
- ❖ The for each Loop

C# Programming

- ❖ The "Hello,World!" C# Program
- ❖ Compiling and Running the Program
- ❖ Comments
- ❖ Class Declaration
- ❖ The Main Method
- ❖ Using the NET Methods for Displaying Results
- ❖ Using Directives
- ❖ Using Local Variables
- ❖ The Program Architecture
- ❖ Qualifying Names
- ❖ Common Conventions for Writing Code
- ❖ Code Documentation

C# Data Types

- ❖ Data Types
- ❖ Builtin Data Types
- ❖ Value Types
- ❖ Variable Initialization
- ❖ Default Values
- ❖ Reference Types
- ❖ The C# Reference Types
- ❖ Boxing and Unboxing
- ❖ Simple Data Types
- ❖ Creating and Manipulating Arithmetic Expressions
- ❖ The Basic Arithmetic Operators (+,-,*,/)
- ❖ The Modulus Operator (%)
- ❖ The Assignment Operators
- ❖ Increment and Decrement Operators (++,—)
- ❖ Operator Associativity
- ❖ How to Get the TypeName
- ❖ Evaluating Expressions with Mixed Types
- ❖ Adding a Suffix to Numeric Data
- ❖ Real Types
- ❖ Integral Types
- ❖ Conversion between Types
- ❖ The char Type
- ❖ Formatting Results
- ❖ The Currency Format
- ❖ The Decimal Format
- ❖ The Fixed point Format
- ❖ The General Format
- ❖ The Numeric Format
- ❖ The Scientific Format
- ❖ The Hexadecimal Format
- ❖ The Nullable Types
- ❖ Using the Nullable Structure Properties
- ❖ Using the ?? Operator
- ❖ The string Type
- ❖ String Expressions
- ❖ String Operators
- ❖ String Concatenation (+,+=)
- ❖ Using the StringBuilder Class
- ❖ The Equality Operator (==)
- ❖ The [] Operator
- ❖ The @ Symbol
- ❖ Reading the Keyboard Input
- ❖ Converting Strings to Numbers
- ❖ Using the Convert
- ❖ Using the Convert
- ❖ Class using the parse Method

UsingClasses

- ❖ Classes
- ❖ ClassDeclaration
- ❖ Field Initialization
- ❖ Class Instantiation
- ❖ Namespaces
- ❖ Nesting Namespaces
- ❖ TheNamespaceAliasQualifier
- ❖ Access Levels
- ❖ Properties
- ❖ UsingProperties
- ❖ ReadonlyProperties
- ❖ Accessor Accessibility
- ❖ StaticMembers andStaticClasses
- ❖ Constants
- ❖ Constructors
- ❖ InstanceConstructors
- ❖ DeclaringConstructors
- ❖ Usingthis
- ❖ PrivateConstructors
- ❖ StaticConstructors
- ❖ ReadonlyFields
- ❖ Inheritance
- ❖ Destructors
- ❖ PartialClasses

FunctionMembers

- ❖ FunctionMembers
- ❖ Polymorphism
- ❖ Virtual and Override Methods
- ❖ Calling Members of the Base Class
- ❖ Overriding Virtual Methods on the Base Class
- ❖ Abstract Classes and Methods
- ❖ MethodOverloading
- ❖ Passing Parameters to Methods
- ❖ Various Ways to Pass Parameters to Methods
- ❖ Usingref
- ❖ Usingout
- ❖ Usingparams
- ❖ Indexers
- ❖ UserdefinedOperators
- ❖ OverridingtheToStringMethod

Structs, Enums,andAttributes

- ❖ StructsvsClasses
- ❖ DeclaringandUsingStructs
- ❖ Passing Structs and Classes to Methods
- ❖ Enumerations
- ❖ DeclaringEnumerations
- ❖ UsingEnumerations
- ❖ Using NET Methods with enums
- ❖ Attributes
- ❖ AttributeParameters
- ❖ TheConditionalAttribute
- ❖ CombiningAttributes
- ❖ Calling Native Functions
- ❖ EmulatingUnions

Generics

- ❖ WhatAreGenerics?
- ❖ UsingGenericCollections
- ❖ List<T>
- ❖ List<T>Members
- ❖ Dictionary<TKey,TValue>
- ❖ Dictionary<TKey,TValue>Members
- ❖ LinkedList<T>

Interfaces

- ❖ What Is an Interface
- ❖ Declaring an Interface
- ❖ Interface Implementation
- ❖ Explicit Interface Implementation
- ❖ Using is to Test Types
- ❖ Using as to Test Types
- ❖ Hiding Members of the Base Class
- ❖ Versioning
- ❖ Hiding Interface Members

Exceptions

- ❖ Errors andExceptions
- ❖ ThrowinganException
- ❖ CatchinganException
- ❖ OrganizingtheHandlers
- ❖ SequenceofEvents inHandlingExceptions
- ❖ ExpectedExceptions inFileProcessing
- ❖ ReadingTextFiles
- ❖ Writing and Appending Text Files
- ❖ ExpectedExceptions
- ❖ The finallyBlock
- ❖ The tryfinallyStatement
- ❖ The trycatchfinallyStatement
- ❖ UserdefinedExceptions
- ❖ RethrowingExceptions
- ❖ RethrowingtheExceptionBacktoMain
- ❖ RethrowingbytheHandlerBlock
- ❖ UsingtheStackTraceProperty

DelegatesandEvents

- ❖ What Is aDelegate?
- ❖ DeclaringDelegates
- ❖ CreatingaDelegate
- ❖ Invoking the Delegate
- ❖ AssociatingaDelegatewithMoreThanOneMethod
- ❖ AddingandRemovingDelegates
- ❖ Using NET Methods to Add and Remove Delegates
- ❖ Anonymous Methods
- ❖ OuterVariables
- ❖ Restrictions on Using Anonymous Methods
- ❖ Covariance
- ❖ Contravariance
- ❖ Events
- ❖ UsingEvents inApplications

CollectionsandIterators

- ❖ CollectionsClasses
- ❖ TheStackCollection
- ❖ StackMembers
- ❖ TheQueueCollection
- ❖ QueueMembers
- ❖ TheArrayListCollection
- ❖ ArrayListMembers
- ❖ TheSortedListCollection
- ❖ SortedListMembers
- ❖ TheHashtableCollection
- ❖ HashtableMembers
- ❖ SpecializedCollections
- ❖ TheListDictionaryCollection
- ❖ ListDictionaryMembers
- ❖ TheLinkedListCollection
- ❖ UsingEnumerators
- ❖ Iterators
- ❖ The IteratorBlocks
- ❖ TheyieldStatement

- ❖ LinkedList<T>Members
- ❖ LinkedListNode<T>Members
- ❖ ICollection<T>
- ❖ ICollectionMembers
- ❖ IDictionary<TKey,TValue>
- ❖ IDictionaryMembers
- ❖ CreatingYourOwnGenericClasses
- ❖ Generic Methods
- ❖ Generic Methods inside Generic Classes
- ❖ Overloading Generic Methods
- ❖ UsingthedefaultKeyword
- ❖ UsingConstraints
- ❖ TypesofConstraints
- ❖ WhentoUseConstraints
- ❖ GenericDelegates
- ❖ Generic Interfaces
- ❖ BenefitsofUsingGenerics
- ❖ LimitationsofUsingGenerics

Building Data Access Components with ADONET

- ❖ Connected Data Access
- ❖ Using the Connection Object
- ❖ Using the Command Object
- ❖ Using the DataReader Object
- ❖ Disconnected Data Access
- ❖ Using the DataAdapter Object
- ❖ Using the DataTable Object
- ❖ Using the DataView Object
- ❖ Using the DataSet Object
- ❖ Executing Asynchronous Database Commands
- ❖ Using Asynchronous ADONET Methods
- ❖ Using Asynchronous ASPNET Pages
- ❖ Building Database Objects with the NET Framework
- ❖ Enabling CLR Integration
- ❖ Creating UserDefined Types with the NET Framework
- ❖ Building a Data Access Layer with a UserDefined Type
- ❖ Creating Stored Procedures with the NET Framework
- ❖ Creating the Stored Procedure Assembly

Assemblies and Versioning

- ❖ PE Files
- ❖ Metadata
- ❖ Security Boundary
- ❖ Versioning
- ❖ Manifests
- ❖ MultiModule Assemblies
- ❖ Private Assemblies
- ❖ Shared Assemblies
- ❖ Public Key Encryption

Attributes and Reflection

- ❖ Attributes
- ❖ Intrinsic Attributes
- ❖ Custom Attributes
- ❖ Reflection
- ❖ Reflection Emit
- ❖ Marshaling and Remoting
- ❖ Application Domains
- ❖ Context

Threads and Synchronization

- ❖ Threads
- ❖ Synchronization
- ❖ Race Conditions and Deadlocks

Streams

- ❖ Files and Directories
- ❖ Reading and Writing Data
- ❖ Asynchronous I/O
- ❖ Network I/O
- ❖ Web Streams
- ❖ Serialization
- ❖ Isolated Storage