The C# Programming Training

Duration: **4 Day**

BILGINC IT Academy
info@bilginc.com
bilginc.com

Overview

Microsoft's .NET Framework presents developers with unprecedented opportunities. From web applications to desktop and mobile platform applications - all can be built with equal ease, using substantially the same skill-set. But to make the most of this potential, developers must have a thorough grasp of language skills and object oriented programming concepts (OOP).

This course concentrates on the C# programming language itself, to fully prepare delegates in readiness for using the .NET Framework/Core. No Object-Orientated knowledge is assumed. From basic procedural syntax to sophisticated object-oriented programming techniques, delegates will learn how to write .NET code that is robust and maintainable.

The course is presented as a mixture of lectures, demos and hands-on exercises. Practical sessions follow all main topics, designed to reinforce the points covered.

This intensive course is intended for developers who will use C# to write .NET Framework or .NET Core applications and who are relatively new to the C# programming language.

Prerequisites

Delegates must understand the fundamentals of programming. Delegates can gain this level of knowledge by attending QA's 'Programming Foundations' (QAPROGFOU) course

Please note:

- Delegates attending this course via Extended Classroom will be required to provide dual monitors.
- Before attending this course, delegates must have a Microsoft account (signing up one is free). The instructions on how to set up a Microsoft account can be found here.

The C# Programming Training

Duration: **4 Day**

BILGINC IT Academy
info@bilginc.com
bilginc.com

What You Will Learn

- Write code that includes sequence, selection and iteration constructs
- Create and use classes and structures (types), including fields, properties, methods and delegates
- Use private, internal, protected and public visibility modifiers
- Use exception handling to create robust applications
- Create derived classes that inherit from custom-written or .NET classes
- Create interfaces and apply techniques of polymorphism effectively and appropriately
- Work with generic types
- Leverage the power of C# features
- Use delegates
- Use lambda expressions in C# code
- Use LINQ (Language Integrated Query)

In addition to the above, the following practice is introduced during the course:

- Test-Driven Development using XUnit

Outline

Module 1: Course Introduction

Module 2: Introduction to C#

- The .NET Framework; Compilation; The Common Language Runtime; The Common Type System
- What is C#?
- Top-level statements
- Namespaces and using directives
- Get to know your way around Visual Studio
- An introduction to debugging
- Start using Unit Test

Module 3: Variables and Datatypes

- Comments and identifiers
- Variables and built-in datatypes
- Value types versus reference types
- Built-in value types: integral, floating point, bool, char, structs and enums
- Reference types: object, strings, object instantiation
- Variable scope and access modifiers
- Operators
- Parsing and casting

Module 4: Conditionals

- If statements
- Switch statements and switch expressions
- Ternary conditional operator
- Null-coalescing and null-conditional operators

Module 5: Loops and Collections

BİLGİNÇ IT Academy

info@bilginc.com

bilginc.com

- Arrays
- Foreach loops
- For loops
- While loops
- Do loops
- Generic collections: List and Dictionary
- Collection operators

Module 6: Object Oriented Programming (OOP)

- Why object orientation?
- OOP concepts and unified modelling language (UML)
- Classes and instantiation
- Association and aggregation
- Inheritance and interfaces

Module 7: Methods

- Methods and positional and named parameters
- Method overloading
- Return values
- Expression-bodied methods
- Passing parameters: by value and by reference
- Static methods and the using static directive
- Extension methods

Module 8: Properties and Constructors

- Fields
- Properties with backing fields
- Expression-bodied properties
- Auto-implemented properties and calculated properties
- How to access properties
- Constructing objects: constructors and expression-bodied constructors
- Constructor overloading and chaining
- Object initializers
- Static constructors
- The ‘this’ keyword

Module 9: Inheritance and Abstract Classes

- Inheritance
- Derived constructors
- Polymorphism
- Virtual members and overriding
- Invoking base class functionality
- Abstract classes
- Upcasting and safe down-casting
- Overriding System.Object
- Sealed classes and members

Module 10: Interfaces

- Creating and implementing interfaces

BILGINC IT Academy

info@bilginc.com

bilginc.com

- Polymorphism
- Interfaces with default implementation
- Interface inheritance
- implementing interfaces explicitly

Module 11: Delegates and Lambdas

- Delegates: Func and Action
- Lambda expressions
- The Predicate delegate

Module 12: Language Integrated Query (LINQ)

- LINQ syntax
- Projections
- Deferred execution and immediate execution
- Joins and aggregations
- The let clause
- The OfType method

Module 13: Exception Handling

- Exception handling keywords: try, catch, finally, throw
- Understanding execution flow
- Throwing exceptions
- Custom exceptions
- Filtered exceptions
- Inner exceptions