The C# Programming Language Training

Duration: **4 Day**

BİLGİNÇ IT Academy
info@bilginc.com
bilginc.com

Microsoft's .NET Framework provides software developers with extensive opportunities to create applications for different platforms. From **web applications** to **desktop and mobile solutions**, developers can build a wide range of software using largely the same programming skills.

To take full advantage of these capabilities, however, developers need a solid understanding of the **C# programming language** and the principles of **Object-Oriented Programming (OOP)**.

The C# Programming Training focuses specifically on C#, preparing participants to develop applications using **.NET Framework** and **.NET Core**.

No previous knowledge of **object-oriented programming** is assumed. The programme begins with fundamental procedural programming concepts and progresses towards more advanced **object-oriented development techniques**.

Throughout the course, learners develop the skills needed to write **robust, maintainable .NET code**.

Training is delivered through a combination of **instructor-led explanations**, **demonstrations** and **hands-on exercises**. Practical activities follow the main topics to reinforce understanding and provide opportunities to apply newly acquired knowledge.

This intensive programme is intended for developers who plan to build **.NET Framework or .NET Core applications using C#** and have relatively limited experience with the language.

Prerequisites

Participants must have an understanding of **fundamental programming concepts**.

Those who need to develop this foundation can do so by attending the **Programming Foundations** course before taking this training.

Important Notes:

- Participants attending through **Extended Classroom** are required to have **dual monitors**.
- Learners must have a **Microsoft account** before attending the course. Creating a Microsoft account is **free of charge**.

Who Should Attend

This course is designed for **software developers who want to learn C#** and use it to create applications with **.NET Framework** or **.NET Core**.

It is particularly suitable for developers who already understand basic programming principles but are **relatively new to C#**.

The C# Programming Language Training

Duration: **4 Day**

BILGINÇ IT Academy
info@bilginc.com
bilginc.com

What You Will Learn

By completing **The C# Programming Training**, participants will gain the knowledge required to use both fundamental and advanced C# features when developing **object-oriented .NET applications**.

By the end of the course, learners will be able to:

- Write code using **sequence**, **selection** and **iteration** constructs.
- Create and work with **classes** and **structures**, including **fields**, **properties**, **methods** and **delegates**.
- Apply **private**, **internal**, **protected** and **public** access modifiers.
- Develop more reliable applications using **exception handling**.
- Create **derived classes** that inherit from custom classes or existing**.NET classes**.
- Build **interfaces** and apply **polymorphism** appropriately and effectively.
- Work with **generic types**.
- Make effective use of the features provided by the **C# language**.
- Use **delegates** in application code.
- Apply **lambda expressions** in C#.
- Write queries using **LINQ (Language Integrated Query)**.

The course also introduces **Test-Driven Development (TDD)** using **xUnit**.

Training Outline

Module 1: Course Introduction

This opening module introduces the course and provides an overview of the C# programming topics to be covered.

Module 2: Introduction to C#

This module introduces the **C# programming language**, the main components of the .NET platform and the development environment.

Topics Covered:

- **The .NET Framework**
- **Compilation**

BİLGİNÇ IT Academy

info@bilginc.com

bilginc.com

- **Common Language Runtime (CLR)**
- **Common Type System (CTS)**
- **What is C#?**
- **Top-level statements**
- **Namespaces** and **using directives**
- Navigating the **Visual Studio** development environment
- Introduction to **debugging**
- Getting started with **unit testing**

Module 3: Variables and Datatypes

This module examines how **variables**, **datatypes** and operators are used in C# applications.

Topics Covered:

- **Comments** and **identifiers**
- **Variables** and built-in datatypes
- Differences between **value types** and **reference types**
- Built-in value types:
  - **Integral types**
  - **Floating-point types**
  - **bool**
  - **char**
  - **structs**
  - **enums**
- Reference types:
  - **object**
  - **strings**
  - **Object instantiation**
- **Variable scope** and **access modifiers**
- **Operators**

BILGINÇ IT Academy

info@bilginc.com

bilginc.com

- **Parsing** and **casting**

Module 4: Conditionals

This module explores the **conditional constructs** used to control program execution based on different conditions.

Topics Covered:

- **if statements**
- **switch statements** and **switch expressions**
- **Ternary conditional operator**
- **Null-coalescing operators**
- **Null-conditional operators**

Module 5: Loops and Collections

Participants learn how to use **loops** for repeated operations and **collections** for organising and managing data.

Topics Covered:

- **Arrays**
- **foreach loops**
- **for loops**
- **while loops**
- **do loops**
- **Generic collections: List** and **Dictionary**
- **Collection operators**

Module 6: Object-Oriented Programming (OOP)

This module introduces the core principles of **Object-Oriented Programming** and demonstrates how they are applied in C#.

Topics Covered:

- **Why use object-oriented programming?**
- **OOP concepts** and **Unified Modeling Language (UML)**
- **Classes** and **instantiation**

BILGINC IT Academy

info@bilginc.com

bilginc.com

- **Association** and **aggregation**
- **Inheritance** and **interfaces**

Module 7: Methods

This module focuses on using **methods** to define application behaviour and create reusable code.

Topics Covered:

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

Module 8: Properties and Constructors

This module examines how **properties** are used to manage class data and how **constructors** support object creation.

Topics Covered:

- **Fields**
- **Properties with backing fields**
- **Expression-bodied properties**
- **Auto-implemented properties**
- **Calculated properties**
- **Accessing properties**
- Creating objects with:
  - **Constructors**
  - **Expression-bodied constructors**
- **Constructor overloading** and **constructor chaining**
- **Object initializers**

BILGINÇ IT Academy

info@bilginc.com

bilginc.com

- **Static constructors**
- The **this** keyword

Module 9: Inheritance and Abstract Classes

This module explores important object-oriented techniques, including**inheritance**, **polymorphism** and **abstract classes**.

Topics Covered:

- **Inheritance**
- **Derived constructors**
- **Polymorphism**
- **Virtual members** and **overriding**
- Calling **base class functionality**
- **Abstract classes**
- **Upcasting** and **safe downcasting**
- **Overriding System.Object**
- **Sealed classes** and **sealed members**

Module 10: Interfaces

Participants explore how to create and implement **interfaces** and how interfaces support polymorphic behaviour.

Topics Covered:

- **Creating and implementing interfaces**
- **Polymorphism**
- **Interfaces with default implementation**
- **Interface inheritance**
- **Explicit interface implementation**

Module 11: Delegates and Lambdas

This module introduces the use of **delegates** and **lambda expressions** in C# programming.

Topics Covered:

- **Delegates**

BILGINC IT Academy

info@bilginc.com

bilginc.com

- **Func** and **Action**
- **Lambda expressions**
- **The Predicate delegate**

Module 12: Language Integrated Query (LINQ)

This module demonstrates how **LINQ** can be used to query and transform data within C# applications.

Topics Covered:

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

Module 13: Exception Handling

The final module focuses on managing runtime errors and improving application reliability through **exception handling techniques**.

Topics Covered:

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

This course enables participants to build practical experience with **C# programming**, progressing from essential language syntax to advanced **object-oriented techniques**, and establishing the skills required to develop **maintainable .NET Framework and .NET Core applications**.