C# Programming Duration – 4 Week

# Overview

# Overview of C# Programming Language

# Environment

# The .Net Framework

# Integrated Development Environment (IDE) for C#

# Creating Hello World Program

# Creating Hello World Program

# Compiling and Executing the Program

# C# Keywords

# Data Types

# Value types

# Reference types

# Pointer types

# Type Conversion

# Implicit type conversion

# Explicit type conversion

# C# Type Conversion Methods

# Variables

# Defining Variables

# Initializing Variables

# Constants

# Character Constants

# Defining Constants

# Operators

# Arithmetic Operators

# Relational Operators

# Logical Operators

# Bitwise Operators

# Assignment Operators

# Misc Operators

# Decision Making

# The ? : Operator

# [if statement](https://www.tutorialspoint.com/csharp/if_statement_in_csharp.htm)

# [if...else statement](https://www.tutorialspoint.com/csharp/if_else_statement_in_csharp.htm)

# [nested if statements](https://www.tutorialspoint.com/csharp/nested_if_statements_in_csharp.htm)

# [switch statement](https://www.tutorialspoint.com/csharp/switch_statement_in_csharp.htm)

# [nested switch statements](https://www.tutorialspoint.com/csharp/nested_switch_statements_in_csharp.htm)

# Loops

# [while loop](https://www.tutorialspoint.com/csharp/csharp_while_loop.htm)

# [for loop](https://www.tutorialspoint.com/csharp/csharp_for_loop.htm)

# [do...while loop](https://www.tutorialspoint.com/csharp/csharp_do_while_loop.htm)

# [nested loops](https://www.tutorialspoint.com/csharp/csharp_nested_loops.htm)

# [break statement](https://www.tutorialspoint.com/csharp/csharp_break_statement.htm)

# [continue statement](https://www.tutorialspoint.com/csharp/csharp_continue_statement.htm)

# Infinite Loop

# Strings

# Arrays

# [Object Oriented Programming (OOPs) in C#](https://dotnettutorials.net/lesson/object-oriented-programming-csharp/)

# [Class and Objects in C#](https://dotnettutorials.net/lesson/class-and-objects-csharp/)

# [Constructors in C#](https://dotnettutorials.net/lesson/constructors-csharp/)

# [Destructor in C#](https://dotnettutorials.net/lesson/destructor-csharp/)

# [Access Specifiers in C#](https://dotnettutorials.net/lesson/access-specifiers-csharp/)

# [Encapsulation in C#](https://dotnettutorials.net/lesson/encapsulation-csharp/)

# [Abstraction in C#](https://dotnettutorials.net/lesson/abstraction-csharp-realtime-example/)

# [Inheritance in C#](https://dotnettutorials.net/lesson/inheritance-c-sharp/)

# [Interface in C#](https://dotnettutorials.net/lesson/interface-c-sharp/)

# [Multiple Inheritance in C#](https://dotnettutorials.net/lesson/multiple-inheritance-csharp/)

# [Abstract Class and Abstract Methods in C#](https://dotnettutorials.net/lesson/abstract-class-abstract-methods-csharp/)

# [Polymorphism in C#](https://dotnettutorials.net/lesson/polymorphism-csharp/)

# [Method Overloading in C#](https://dotnettutorials.net/lesson/function-overloading-csharp/)

# [Method Overriding in C#](https://dotnettutorials.net/lesson/function-overriding-csharp/)

# [Method Hiding in C#](https://dotnettutorials.net/lesson/function-hiding-csharp/)

# [Partial Classes and Partial Methods in C#](https://dotnettutorials.net/lesson/partial-classes-partial-methods-csharp/)

# [Sealed Class in C#](https://dotnettutorials.net/lesson/sealed-class-methods-csharp/)

# [Extension Methods in C#](https://dotnettutorials.net/lesson/extension-methods-csharp/)