**.NET CORE FRAMEWORK 8, ASP.NET CORE, EF CORE WITH C# 12**

## Duration:

10 days (Suggested)

## Pre-requisites:

Participants should have knowledge in Object-oriented programming. If they know any of the following languages, that will be additional benefit: C++, Java etc.

Participants should have knowledge in HTML, CSS, Bootstrap

Day-1: [6th Jan,2-25]

## Module 1: Overview of .net core framework

* What is the .NET Core/.NET?
* .NET Core Chronology
* .NET Core Architecture
* .NET Library
* .NET Core Runtime, ASP.NET Core Runtime
* Just-In-Time (JIT) compiler
* Common Type System (CTS)
* Common Language Specifications (CLS)
* Benefits of .NET:
  + Cross-Language Interoperability
* Assemblies
* Microsoft Intermediate Language (MSIL) or CIL (Common Language Runtime)
* Overview of .NET Core and its versions since 2016

## Module 2: editor: Visual Studio IDE

* Visual Studio 2019/2022
* Applications You Can Write in VS IDE
* Different Types of Project Templates
* Solution and Project Architecture in VS IDE Applications
* “bin” and “Debug/Release” Folder
* Different Windows i.e. Solution Explorer, Properties, Server Explore, etc. in Visual Studio

## Module 3: Introduction to C# Programming Language and Constructs of a C# Application

* What is C#?
* History and Chronology of C#
* Features of C#
* Basic C# Syntax
* Basic C# Console Application Structure
* Main Method
* Namespaces
* “using” directive
* Console I/O
* Comments
* Code comments and documentation comments
* Separation of code
* C# Programming Guidelines
* Hands-on Lab

## Module 4: Data Types, Variables, Operators and Control Flow in C#

* Variables
* Data Types in C#
* Value and Reference Types
* Variable Naming
* Literal Values
* Operators
  + Different types of operators in C#
    - Mathematical Operators
    - Assignment Operators
    - Conditional Operators
    - Boolean Bitwise and Assignment Operators
    - Unary, Binary and Ternary Operator
  + Operator Shortcuts
  + Operator Precedence
* Branching
* The if and if…else Statement
* The switch Statement
* Looping
  + do Loop
  + while Loop
  + for Loop
  + Interrupting Loop
* Hands-On Lab

Day-2: [7th Jan, 2025]

## Module 5: Data Type Conversions and Other Data Types in C#

* Type Conversion
  + Implicit Conversions
  + Explicit Conversions
  + Explicit Conversions Using the Convert class methods
  + Parse method for conversion
  + Boxing and Unboxing
  + Casting
* Complex Variable Types
  + Enumerations
  + Structs
  + Arrays
* Hands-On Lab

## Module 6: Functions

* Defining and Using Functions
* Return Values
* Parameters
* Variable Scope
* Variable Scope in Other Structures
* Parameters and Return Values
* The Main () Function
* Hands-On Lab

## Module 7: Arrays

* What is an Array?
* The System.Array Class
* Using an Array
* Types of Arrays
* Single-Dimensional Array
* Multi-Dimensional Array

Day-3: [8th Jan, 2025]

## Module 8: OOP-Defining Classes, Abstraction and Encapsulation

* Classes in C#
* How to declare in class in C#
* Create an object in C#
* Constructors
* Hands-On Lab
* Understanding and Implementing Data Abstraction
* Understanding and Implementing Encapsulation
* Using Properties in C# for data encapsulation
* Hands-On Lab

## Module 9: OOP-Inheritance

* Inheritance in C#
* Types of Inheritance in C#
* Constructor Execution Sequence
* Hands-On Lab

Day-4: [9th Jan, 2025]

## Module 10: OOP-Polymorphism

* Polymorphism in C#
* Types of Polymorphism:
  + Static/Compile Time polymorphism
    - Method Overloading
    - Operator Overloading
* Dynamic/Runtime polymorphism
  + Override and Virtual keyword
* Difference between Static and Dynamic Polymorphism
* Hands-On Lab

## Module 11: OOP-Abstract Class, Interface, design patterns

* Abstract Class
* Implementing Abstract Class
* Interface
* Implementing Interface
* Difference between an Abstract Class and an Interface
* Hands-On Lab

Day-5: [10th Jan, 2025]

## Module 12: OVERRIdiNG Object class METHODS in C#

* Object class and its methods
* Equals method
  + Comparing Reference Types of Equality
  + Comparing Value Types of Equality
* GetHashCode() method
* GetType() method
* ToString() Method
* “is” and “as” operator
* Hands-On Lab

## Module 13: design patterns and architecture

* SOLID Principles
* IoC and Dependency Injection
* Design Patterns in .NET: Singleton, Factory, Builder etc.

Day-6: [13th Jan, 2025]

## Module 13: Errors and Exception Handling

* Introduction to Errors and Exceptions
* The System.Exception Class
* Exception Classes
* Try…Catch…Finally Blocks
* Nested Try Blocks
* Catching Exceptions
* Implementing Multiple Catch Blocks
* Catching Exceptions from Other Code
* Throw Statement
* User-Defined Exception Classes
* Hands-On Lab

## Module 14: Generics

* Introduction to Generics
* Benefits of Generics
* Generic Type Parameter
* Using Generics
* Nullable Types
* Defining Generic Types
  + Defining Generic Classes
  + Defining Generic Interfaces
  + Defining Generic Methods
* Default Keyword
* Constraints on Type Parameters
* Hands-On Lab

Day-7: [15th Jan, 2025]

## Module 15: Generic Collections

* What is generic collection class and their advantages
* Introduction to various generic collection types – interface and classes
* The System.Collections.Generic Namespace
* Generic Collection interfaces: ICollection<T>, IEnumerable<T> and IEnumerator<T> Interfaces
* Generics Collections classes:
  + List<T>
  + Queue<T>
  + SortedList<T>
  + Stack<T>
  + SortedSet<T>
  + LinkedList<T>
  + Dictionary<TKey, TValue>
* Hands-on Lab

## Module 16: Delegates

* What is reference to method?
* What is delegate and real-time use
* Declaring Delegates
* Using Delegates to refer method(s)
* Using delegate to invoke method(s)
* Generic delegates
* Action<T>, Predicate<T>, Func<T> Delegates
* Anonymous Methods
* Lambda Expressions
* Hands-On Lab

Day-8: [16th Jan, 2025]

## Module 17: LINQ

* What is LINQ?
* LINQ syntax: Method Query syntax and Query Operator syntax
* LINQ Query Operator Syntax
  + Declaring a Variable for Results Using the var Keyword
  + Specifying the Data Source: from Clause
  + Specify Condition: where Clause
  + Selecting Items: select Clause
  + Finishing Up: Using the foreach Loop
  + Deferred Query Execution
* LINQ Method Syntax
  + LINQ Extension Methods
  + Query Syntax versus Method Syntax
  + Lambda Expressions
* Ordering Query Results
* Understanding the order by Clause
* Querying a Large Data Set
* Using Aggregate Operators

Day-9: [17th Jan, 2025]

## Module 18: File hndling in C#

* System.IO namespace
* File Classes for Input and Output
  + The File and Directory Classes
  + The FileInfo Class
  + The DirectoryInfo Class
  + Path Names and Relative PathsFile Classes for Input and Output
  + The File and Directory Classes
  + The FileInfo Class
  + The DirectoryInfo Class
  + Path Names and Relative Paths
* Streams
  + Classes for Using Streams
  + The FileStream Object
  + The StreamWriter Object
  + The StreamReader Object
  + Asynchronous File Access
* Reading and Writing Compressed Files
* Hands-On Lab

## Module 19: Serialization

* What is serialization?
* Different types of serialization in C#
* Binary, SOAP, XML and JSON Serialization
* Hands-On Lab

Day-10: [20th Jan, 2025]

## Module 20: Threading and Tasks in C#

* Threads in C#
  + Multithreading
  + Thread class
  + Create a thread and Start a Thread
  + Synchronization of thread using Thread.Join, Lock and Monitor
* Tasks
  + Starting Task and Results from Tasks
  + Continuing Tasks
  + Parallel Task Library

## Module 21: Data Access using ADO.NET

## Note: THIS MODULE WILL BE TAUGHT BRIEFLY SO THAT LEARNERS CAN UNDERSTAND THE NEXT MODULE (EF CORE)

* Connection class
* Command class
* Direct Command execution against database
* Using Parameters in command
* Performing CRUD operations

Day-11: [21st Jan 2025]

## Module 22: Entity Framework core

* EF core – introduction
* Installation
* Existing Database
* DbContext
* Creating first application
* Querying
* Saving data
* Conventions
  + One to one
  + One to Many
* Configurations
* Fluent API
  + One to many
  + One to one
  + Many to many
* Working with stored procedure
* Logging
* Migration
  + PMC commands
  + CLI commands

Day-12: [22nd Jan, 2025]

## Module 23: ASP.NET Core Application Architecture

* What is ASP.NET Core and its advantages
* Create an ASP.NET Core application from scratch
  + In Visual Studio Empty project template will be used
* Are we creating console or web application?
* Hosting Environment
  + WebApplication
  + IApplicationBuilder
  + WebHost
  + Which web server is hosting?
  + Kestrel, IIS Express, IIS, Apache etc.

## Module 24: Understanding services and Middleware

* What is IoC Container and Dependency Injection?
* ServiceCollection – container for ASP.NET Core applications
* How to register services
  + DbContext service
  + MVC services
* Introduction to Middleware
* Why middleware?
* Request Processing Pipeline
  + Request Processing pipeline in ASP.NET
* Different middleware
  + For Exception handling
  + For Static Files
  + For environment files
* Creating Custom middleware
  + Conventional way of creating middleware
  + Factory pattern to create middleware
  + Services and Dependency Injection

Day-13: [23rd Jan, 2025]

## Module 25: Application Configuration and Setting files

* Configuration Providers and Sources
* Configuration API
* Options Pattern
* Setting files
  + Launchsetting
  + Appsetting
* Order of reading data from those files

## module 26: .NET Core Services: Logging

* Introduction
* Configuration
* ILogger interface

Day-14: [24th Jan, 2025]

## Module 27: ASP.NET MVC Core – Web Applications

* Introduction to Razor application
* Introduction to MVC application
* Introduction to Web API application
* Creating all the above from scratch using Visual Studio Code
* Creating all the above in Visual Studio using respective templates
* Services, Middleware required for all the above

## module 28: Request Routing

* Routing Middleware
* Route Templates
* Route Constraints
* MVC Middleware
* Attribute-Based Routing
* Conventional Routing
* Why UseRouting() and UseEndpoints() method introduced in ASP.NET Core 3.0?
* Details and emergence of endpoint routing

Day-15: [27th Jan, 2025]

## module 29: Controllers and Views

* What is ViewImports file in ASP.NET Core Razor and MVC applications?
* What is TagHelper and advantages over HtmlHelper
  + Image Tag helper
  + Environment Tag helper
  + Form Tag helper
* Using sections, layout in ASP.NET MVC Core
* Using Bootstrap

## module 30: Error Handling

* Best Practices
* HTTP Error Status Codes
* Status Code Pages
* Developer Exception Page
* Exceptions
* Filters

Day-16: [28th Jan, 2025]

## module 31: State Management

* State management in ASP.NET Core
  + Cookies
  + Session state
  + Temp data
  + ViewData
  + ViewBag
  + Query strings
  + Hidden fields
  + Cache

## module 32: Models

* Introduction
* Persistence Ignorance
* Object-Relational Mapping
* Entity Framework (EF) Core
* Accessing data using EF Core
* Model Validation

Day-17: [29th Jan, 2025]

## Module 33: ASP.NET CORE Web API

* Introduction
* What is RESTful application?
* What is Web API application?
* Difference between Web Service and Web API application
* Middlewares for Web API application
* ASP.NET Core Web API application structure

## Module 34: routing in asp.net core web api

* Routing in Web API
* Routing and Action Selection in Web API
* Attribute Routing in Web API
* Create a REST API with Attribute Routing in Web API

Day-18: [30th Jan, 2025]

## Module 35: Creating a Web API application for CRUD operation

* Retrieval Operations
* Create Operations
* Update Operations
* Delete Operations
* Bad Requests
* How to use Async Task
* Cross-Origin Resource Sharing
* Postman to test API methods

Day-19: [31st Jan, 2025]

## Module 36: Exception Handling and Testing

* Exception Handling in Web API
* Global Error Handling in Web API
* Tracing in Web API
* Unit Testing Web API
* Unit Testing Controllers in Web API

Day-20: [6th Feb, 2025]

## Module 37: Securing Application

* Authentication
* Authorization
* Token Based Authentication
* API Authentication / Authorization
* Cookie Middleware

## Module 38: Deployment and Version Control

* Using Git for version control
* Deploying applications to Azure or other cloud services