

Introduction to ASP.NET• .NET and ASP.NET frameworks:

.NET: .NET is a free, open-source development platform for building many kinds of apps, such as: Web application, Web services, Desktop application, mobile application, game application etc. .NET is a software framework which is designed and developed by Microsoft. On one side, it is a developer platform, on other side, it is an essential component of the operating system that helps in executing applications. There is a variety of programming languages available on the .Net platform, VB.Net and C# being the most common ones.

ASP.NET: ASP.NET is a free web framework designed and developed by Microsoft for building great websites and web applications using HTML, CSS and JavaScript. It is a subset of the .NET framework and successor of the classic ASP (Active Server Pages).

.NET vs ASP.NET

.NET	ASP.NET
i) .NET is a software development framework aimed to develop Windows, Web and Server based applications.	i) ASP.NET is a main tool that present in the .NET framework and aimed at simplifying the creation of dynamic webpages.
ii) Server side and client side application development can be done using .NET framework.	ii) We can only develop server side web applications using ASP.NET as it is integrated with .NET framework.
iii) Mainly used to make business applications on the Windows platform.	iii) Mainly used to make dynamic web pages and websites using .NET languages.
iv) Its programming can be done using any language with CIL (Common Intermediate language) compiler.	iv) Its programming can be done using any .NET compliant language.

**.NET Core:** .NET Core is a new version of .NET framework. It is general-purpose development platform maintained by Microsoft. It is a cross-platform framework that runs on Windows, macOS, and Linux operating systems, used to build different types of applications such as mobile, desktop, web, cloud, IoT, machine learning, game etc. .NET Core is written from scratch to make it modular, lightweight, fast, and cross-platform Framework.

### .NET Core Characteristics:

i) Open-source framework: .NET Core is an open-source framework maintained by Microsoft and available on GitHub under MIT and Apache 2 licenses.

ii) Cross-platform: .NET Core runs on Windows, macOS, and Linux operating systems. There are different runtime for each operating system that executes the code and generates the same output.

iii) Consistent across Architectures: Execute the code with the same behaviour on different instruction set architectures, including x64, x86, and ARM.

iv) Wide-range of Applications: Various types of applications can be developed and run on .NET Core platform such as mobile, desktop, web, cloud, IoT, games etc.

v) Supports Multiple Languages: We can use C#, F#, and Visual Basic programming languages to develop .NET Core applications.

vi) Compatibility: Compatible with .NET Framework and Mono APIs by using .NET Standard Specification.

**Mono:** Mono is an example of a cross-platform framework available on Windows, macOS, Linux and more. It was first designed as an open source implementation of the .NET Framework on Linux. Mono is known for its high level of portability. For example, the Unity game engine uses C# as a cross-platform way of creating video games. Mono provides the means to compile, and run C# programs, similar to the .NET Framework.

## ⑧ ASP .NET WEB FORMS:

It is a part of the ASP.NET web application framework and is included with Visual Studio. Web Forms are pages that users request using their browser. These pages can be written using a combination of HTML, client-script, server controls, and server code. When users request a page, it is compiled and executed on the server by the framework, and then the framework generates the HTML markup that the browser can render. An ASP.NET Web Forms page presents information to the user in any browser or client device.

### Features of ASP.NET WEB FORMS :

- ⇒ Server Controls: ASP.NET Web server controls are similar to familiar HTML elements, such as buttons and text boxes. Other controls are calendar controls, and controls that we can use to connect to data sources and display data.
- ⇒ Master Pages: ASP.NET master pages allow us to create a consistent layout for the pages in our application. A single master page defines the look and feel and standard behaviour for all of the pages in our application.
- ⇒ Working with Data: ASP.NET provides many options for storing, retrieving and displaying data in web page UI elements such as tables and text boxes and drop-down lists.
- ⇒ State Management: ASP.NET Web Forms includes several options that help us preserve data on both a per-page basis and an application-wide basis.
- ⇒ Security: offer features to develop secure application from various security threats.
- ⇒ Performance: offers performance related to page and server control processing, state management, data access, application configuration and loading, and efficient coding practices.

## ④ ASP .NET MVC:

It is the Model-View-Controller application model which can be merged with the new ASP.NET Core. It is used to build dynamic websites as it provides fast development. ASP.NET MVC gives us a powerful, patterns-based way to build dynamic websites that enables a clean separation of concerns and that gives us full control over markup for enjoyable, agile development.

## ⑤ ASP .NET Web API:

also include MVC pattern/architecture from unit 3 if asked in exam.  
i.e, describe model, view and controller.

It is the Web Application Programming Interface (API). ASP.NET Web API is a framework that makes it easy to build HTTP services that reach a broad range of clients, including browsers and mobile devices. ASP.NET Web API is an ideal platform for building RESTful applications on the .NET Framework. The ASP.NET Web API is an extensible framework for building HTTP based services that can be accessed in different application on different platform such as web, windows, mobile etc. It is more or less similar to that of ASP.NET MVC except that it sends data as response of HTML view.

## ⑥ .NET Architecture and Design Principles:

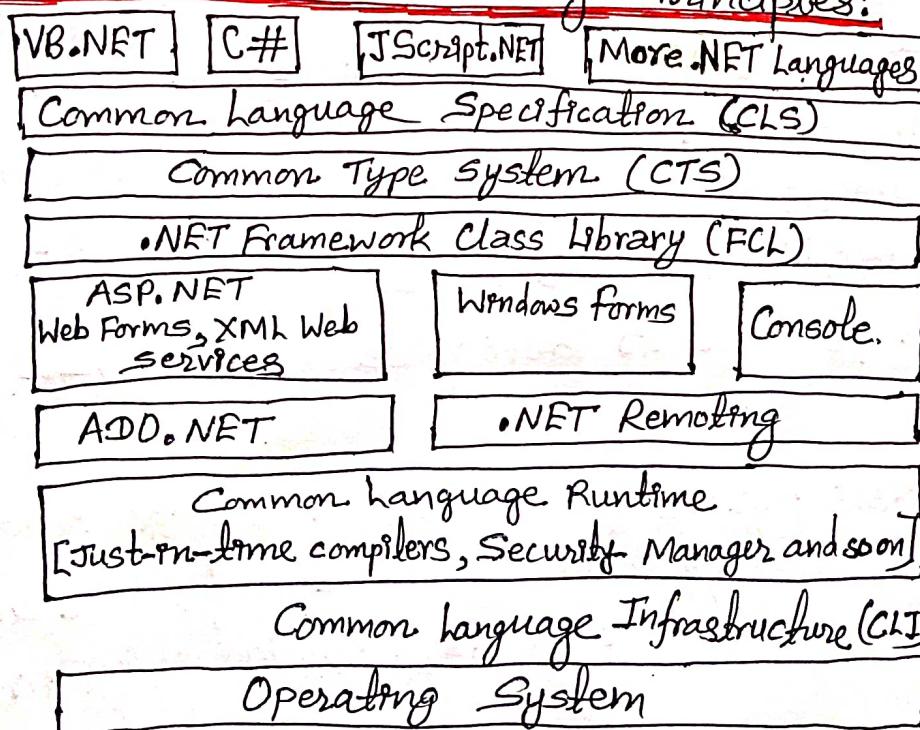


Fig: An overview of .NET architecture.

## ④ ASP .NET MVC:

It is the Model-View-Controller application model which can be merged with the new ASP.NET Core. It is used to build dynamic websites as it provides fast development. ASP.NET MVC gives us a powerful, patterns-based way to build dynamic websites that enables a clean separation of concerns and that gives us full control over markup for enjoyable, agile development.

## ⑤ ASP .NET Web API:

also include MVC pattern/architecture from unit 3 if asked in exam.  
i.e, describe model, view and controller.

It is the Web Application Programming Interface (API). ASP.NET Web API is a framework that makes it easy to build HTTP services that reach a broad range of clients, including browsers and mobile devices. ASP.NET Web API is an ideal platform for building RESTful applications on the .NET Framework. The ASP.NET Web API is an extensible framework for building HTTP based services that can be accessed in different application on different platform such as web, windows, mobile etc. It is more or less similar to that of ASP.NET MVC except that it sends data as response of HTML view.

## ⑥ .NET Architecture and Design Principles:

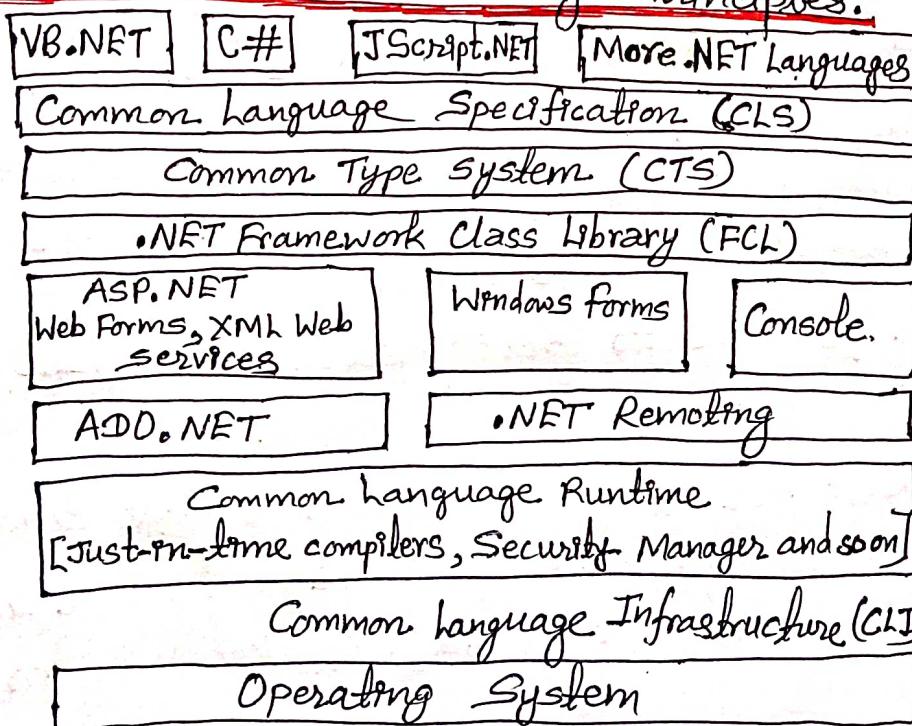


Fig: An overview of .NET architecture.

Common Language Runtime (CLR): It is the heart of .NET Framework. It resides above the operating system and handles all .NET applications. It handles garbage collection, Code Access Security (CAS) etc.

Common Language Infrastructure (CLI): It provides a language-independent platform for app development and its performance. It also includes the function for exception handling, waste collection, security etc.

Common Type System (CTS): It specifies a standard that represent what type of data and value can be defined and managed in computer memory at runtime.

Common Language Specification (CLS): It is a subset of CTS that defines a set of rules and regulations which should be followed by every language under the .net framework.

Framework Class Library (FCL): It provides the various system functionality in the .NET framework, that includes, classes, interfaces, data types etc. to create multiple functions and different types of application such as desktop, web, mobile application etc.

### Design Principles of .NET:

- Interoperability
- Portability
- In-built security mechanism.
- Robust memory management.
- Simplified deployment.
- Asynchronous Programming.
- High Performance.

### ④ Compilation and Execution of .NET applications:

(CLI, MSIL and CLR)

Code that is executed by the CLR is sometimes referred to as "managed code", in contrast to "unmanaged code" which is compiled into native machine language that targets a specific system.

language interoperability is a key feature of the .NET framework. Because the IL code produced by the C# compiler conforms to the Common Type Specification (CTS), IL code generated from C# can interact with code that was generated from the .NET versions of Visual Basic, Visual C++, or any of more than 20 other CTS-compliant languages. A single assembly may contain multiple modules written in different .NET Languages, and the types can reference each other just as if they were written in the same language.

## ⑧ .NET CLI: build, run, test and deploy .NET Core Applications:

- The .NET Core command-line interface (CLI) is a new cross-platform tool for creating, restoring packages, building, running and publishing .NET applications.
- Visual Studio internally uses this CLI to restore, build and publish an application. Other higher level IDEs, editors and tools can use CLI to support .NET Core applications.
- The .NET Core CLI is installed with .NET Core SDK for selected platforms. So we don't need to install it separately on the development machine. We can verify whether the CLI is installed properly by opening command prompt in Windows and writing dotnet and pressing Enter. If it displays usage and help then it means it is installed properly.

## Creating and running the Hello World console application:

Execute the following commands on the terminal:

- mkdir hwapp
- cd hwapp
- dotnet new console

The command "dotnet new console" creates a new Hello World console application in the current folder. It creates two files:

- Program.cs
- hwapp.csproj

Program.cs should look similar to following listing:

```
using System;
namespace hwapp
{
    public class Program
    {
        public static void Main (string[] args)
        {
            Console.WriteLine ("Hello World");
        }
    }
}
```

### Running the Hello World console application:

- When we are using .NET Core SDK, our application will be built automatically when needed. There's no need to worry about whether or not we are executing the latest code.
- Try running the Hello World application by executing dotnet run at the terminal or command line.