**Basics of C#.NET Programming**

**(Part no. 1)**

# **Introduction of C#**

C#, usually pronounced or spoken as ‘See-sharp’ or ‘C-Sharp’, is an evolution of the C, C++ programming languages. It was founded by Anders Hejlsberg under Microsoft. It is far more Object-oriented, Component-oriented, Block-based, Structured, and Scalable programming language than its predecessors. Widely used in the development of Desktop applications, Mobile Applications, Web Applications, Game Development, Enterprise Applications, and the list goes on. Now let's move on to its amazing features where we will dissect this language further and get to know more about it.

# **Features**

Let's discuss some of the main and important features of the C# programming language. Their further description is described in the following:

## **Simplicity**

Developing applications in C# is much simpler, block-based, and structured (breaking code at different points or intervals). The syntax is simpler than that of C++ as you don’t have to define and code every pointer or click now.

## **Modern**

C# fulfills the modern demands of the modern era. It can build more scalable, powerful, and secure applications and also provides enterprise solutions for the companies too.

## **OOP based**

C# is a more Object-oriented programming-based language. Which makes it easier to used and much more simple to understand and develop applications. Even there are some Java developers that I know are switching towards this modern language to increase their productivity and provide more scalable options to their clients/users.

## **Uses .NET Libraries**

Yes, you are getting it right. C# is also one of the programming languages designed to work with Microsoft’s .NET platform. This platform is enriched with tons of libraries that you will see, which we will be using in our later tutorials in the future. Almost all the desktop and some Web, Mobile applications are developed using the .NET Framework and .NET libraries with the help of C# as a programming language. Furthermore, about .NET will be discussed in the .NET Framework section. But for the features let's end it here.

## **Speed**

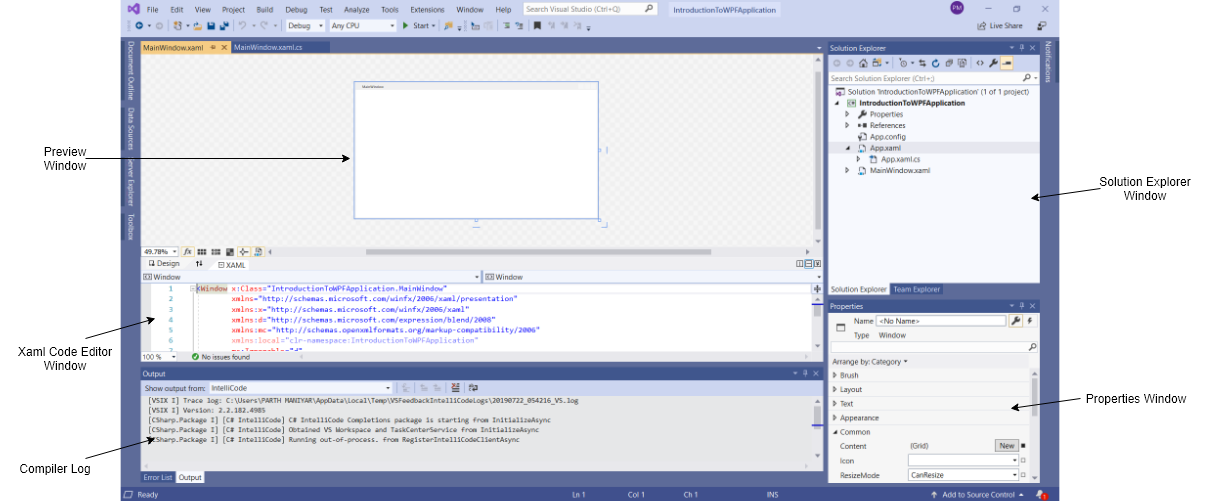
Applications developed in the C# takes less time in the compilation and execution of the software. Thanks to the compilers used in .NET, reducing the time for debugging the software and helping to load it faster on Desktop.

# **Applications in C#**

Some applications that can be written in the C# using .NET are described below

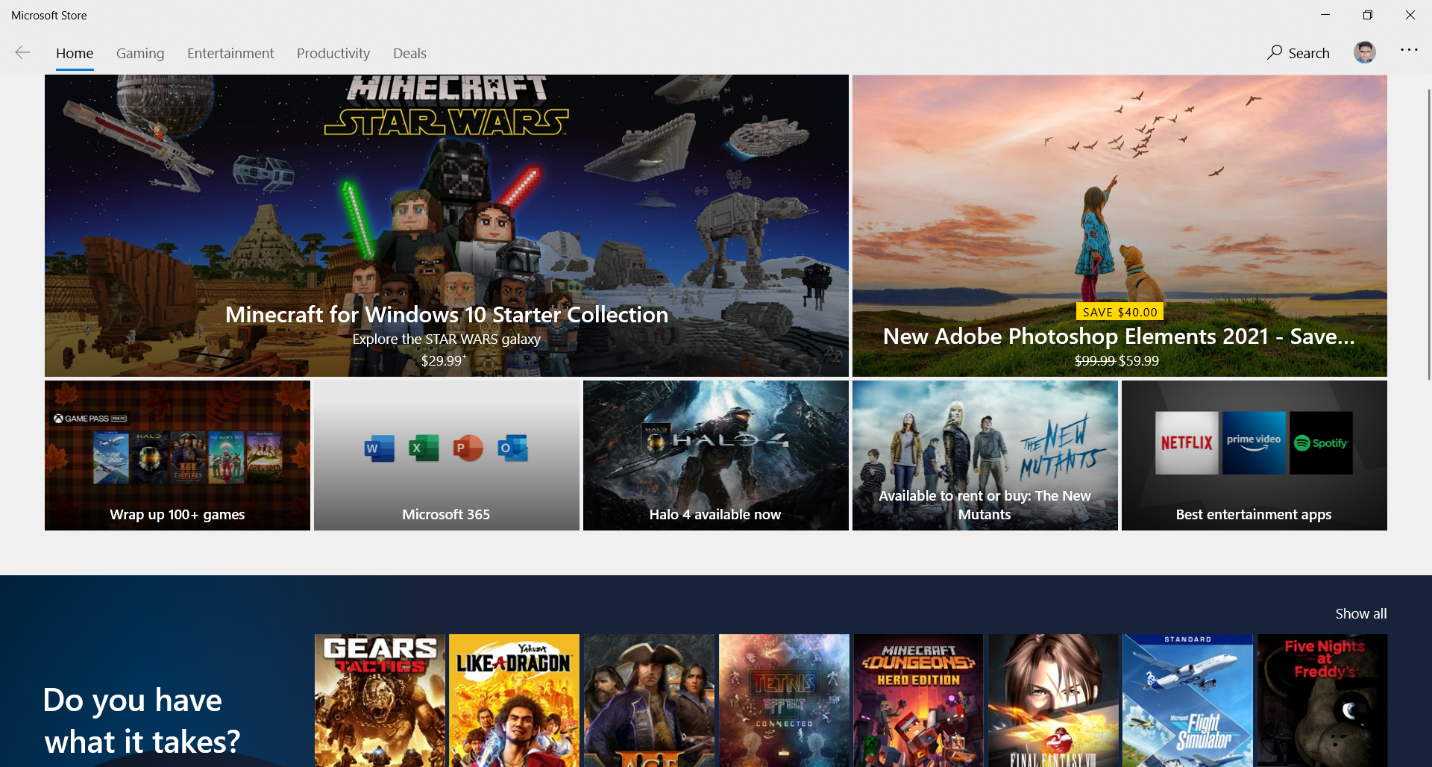
## **Desktop Applications**

For the development of the Desktop application, we will be using the Windows Presentation Foundation also known as the WPF, it is one of the modules of the .NET Framework. Its libraries include buttons, toolbars, text boxes, and so on, which could be used to build a windows user interface or UI.



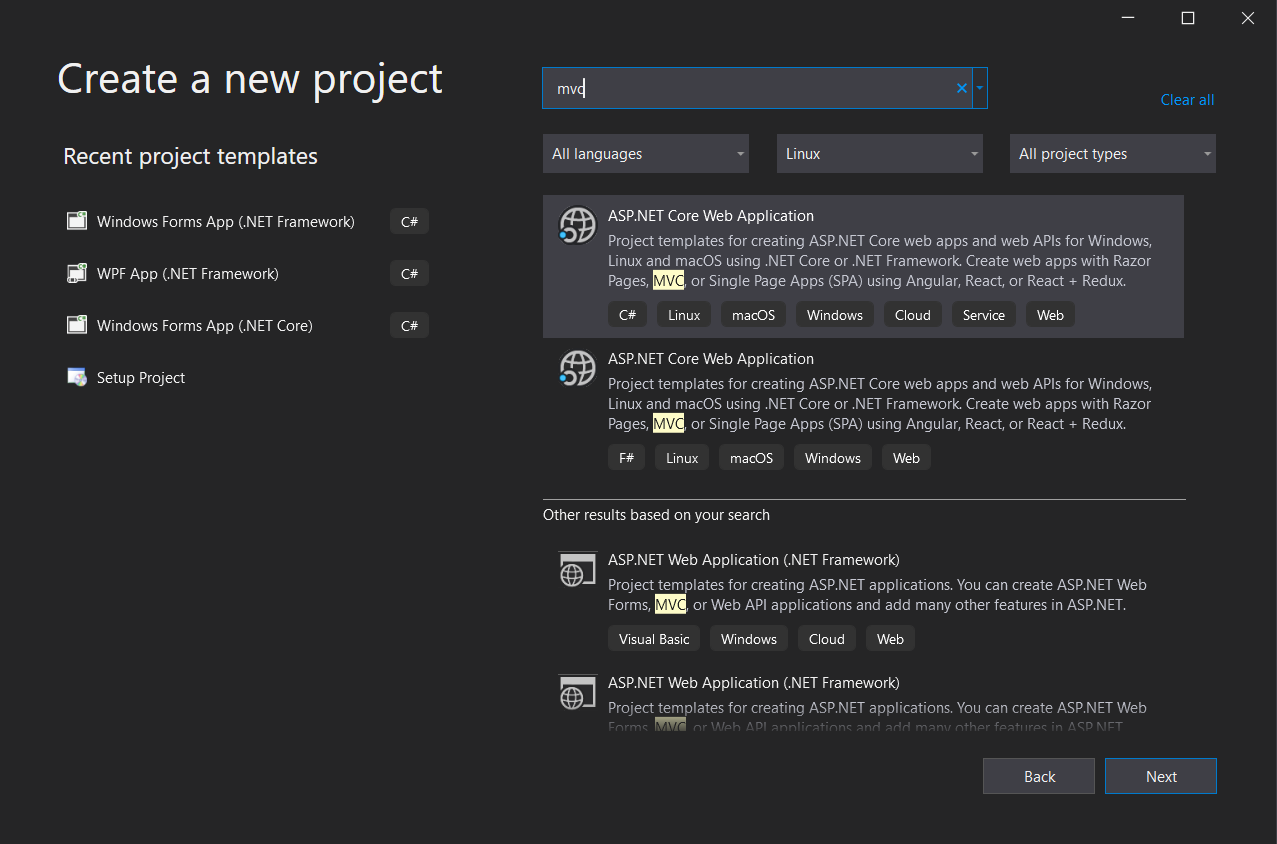
## **Windows Store Applications**

This module is mainly used to develop the applications for the touch devices with a minimum or not at all clutter and keeping the simplicity. WPF is also used to some extent but mostly WCF is used for more scalable apps.



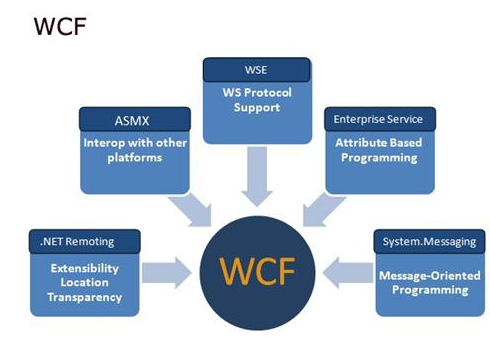
## **Web Applications**

For the Web Application development, we use the ASP.NET (Active Server Page. NET) system. It provides more dynamic, personalized, security to web applications. For this one, we will use Model View Controller, MVC, another module of the .NET



## **WCF Service**

These Services are used to create versatile distributed applications. It’s an abbreviation of Windows Communication Foundation Service. Mostly used for the exchange of the data virtually over the internet or local internet, we will use it in our advance part of the tutorials. Some of these types also need some database access, for that purpose, we will be using the ADO.NET (Active Directory Objects .NET) and LINQ (Language Integrated Query) also one of the capabilities of C#.



# **.NET Framework**

.NET is the abbreviation of Network Enabled Technology. It was the platform developed by Microsoft for developing different applications. An application using .NET Framework means writing code using a .NET code library. So, it consists of a gigantic set of libraries and codes used by the client language such as C# using OOP (Object-Oriented Programming) techniques. It is further categorized into different modules which we discussed in the C# section later. There are some compilers too used by .NET Framework to help in the compilation and execution of the software.

## **Compilers**

When you compile your code using the .NET Framework library, you don’t create an operating system-specific native code, instead, your code is compiled into the Common Intermediate Language (CIL compiler) code which is easier for the operating system to read the code and perform further operations on it. It is neither specific to an operating system nor the C# itself. After that comes the execution part for which Just in time (JIT compiler), it compiles the CIL code to its native code that is specific to the target Operating System. Gosh, that was too much of the compilers. Let’s move to the next step and discuss some features. NET.

# **Features of .NET Framework**

.NET Framework offers a wide variety of features for its users and developers. Let's shed some light on all of them one by one.

## **Assemblies**

When compiling an application, the CIL code is stored in an assembly. Assemblies include both executable code (.exe file extension), application files, and libraries (.dll extension) for being used by other applications. Containing the CIL, assemblies also include the meta-information (information of information about the assemblies). This meta-information enables assemblies to be fully self-descriptive. There is also another useful feature known as Global Assembly Cache (GAC) in the .NET Framework. In this one, all the reusable code is placed inside a place where it could be accessible to all the other applications.

## **More Managed Code**

Code written C# using the .NET Framework is managed when it is executed (a stage usually referred to as runtime). So here the CLR (Common Language Runtime) looks after the applications, managing the memory, handling security, allowing cross-language debugging (using XAMARIN platform or the other), and so on. In short, applications that do not run under the control of the CLR are said to be unmanaged such as C++ which access low-level functions of the operating system. However, C# only runs in a managed environment.

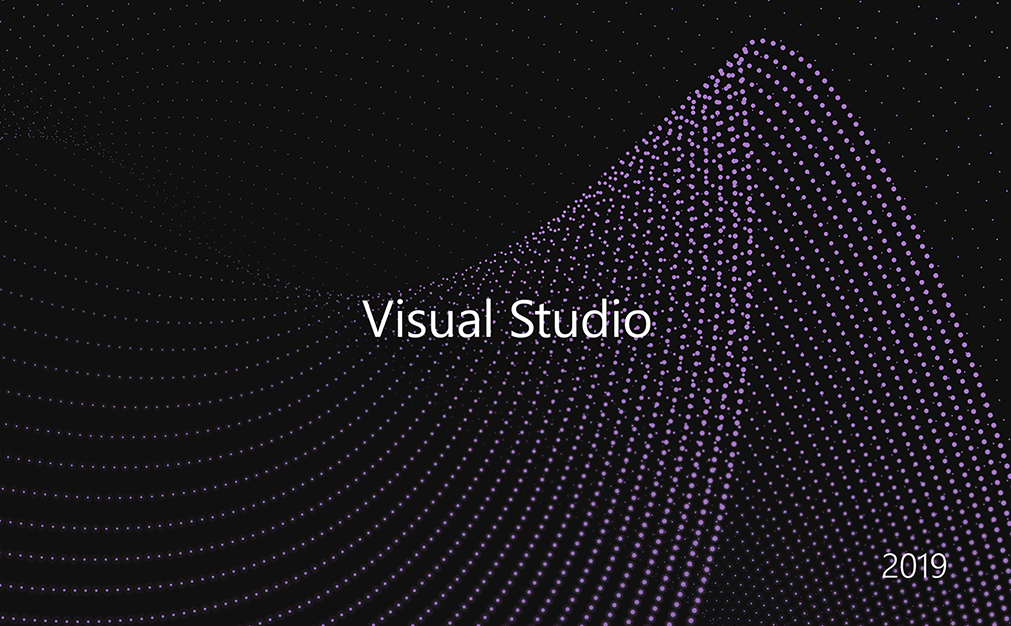
## **Garbage Collection**

An important feature of the .NET Framework in managed code is Garbage Collection. .NET Garbage Collection works by periodically inspecting the memory of the computer and removing anything from it that is no longer needed. It is not set or fixed time for this process, it might happen thousands of times per second, once every few seconds, or whenever, but be assured that it will happen.



# **Integrated Development Environment (IDE)**

For C#.NET we will be using Visual Studio Community 2019 version. This is the best IDE developed by Microsoft for various kinds of single-level or cross-platform applications. Why we are using the 2019 version rather than some old 2015 or 2017 version? Because the 2019 version has more optimization of source codes and files, much more software development features making it available for the different kinds of developers and also introduced various cross-platform integration of different applications. Let's talk about some of its features.



## **Features of Visual Studio**

Some features of the visual studio are shortlisted below which makes it an appealing choice for the .NET development:

* Visual Studio automates the steps required to compile source code but at the same time gives complete control over any options one wants.
* The visual Studio text editor is tailored for the languages that VS supports, including C#. In this way, it can intelligently detect errors and suggest code wherever appropriate as one is typing. This feature is also called ‘IntelliSense’.
* C# projects often start with ‘boilerplate’ code already in place. So instead of starting from scratch, various code files are started, reducing the amount of time spent starting the project.
* It can also add appropriate code to existing files without having you worried about correct syntax.
* Visual Studio also includes various powerful tools for visualizing and navigating through the elements of the projects, whether they are bitmap images or sound files.
* You can also create deployment projects, making it easy for supplying the code to clients and for them to install it without any or much trouble.
* Visual Studio also enables advanced debugging techniques when developing the projects, such as stepping through the code one step at a time. While keeping an eye on the state of the application.

Believe me guys, there is much more than this to discuss, but you get the idea right now! Right? 🤓

I hope you have enjoyed learning some new stuff today, stay tuned as there are furthermore to come. 😎