Visual Basic.Net

(Introduction to .NET Framework)

For

BCA Students

ANANT KUMAR
Faculty Member
Department of Computer Science
J. D. Women's College, Patna

Introduction to .NET Framework

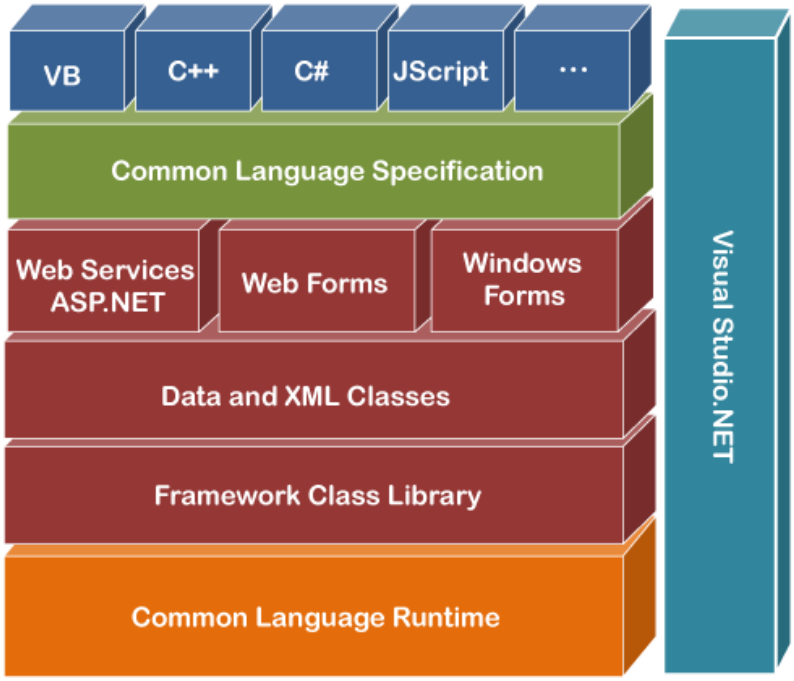
The **.NET Framework** is a software development platform that was introduced by Microsoft in the late 1990. On 13 February 2002, Microsoft launched the first version of the .NET Framework, referred to as the **.NET Framework 1.0**.

What is .NET Framework?

It is a virtual machine that provide a common platform to run an application that was built using the different language such as C#, VB.NET, Visual Basic, etc. It is also used to create a form based, console-based, mobile and web-based application or services that are available in Microsoft environment. Furthermore, the .NET framework is a pure object oriented, that similar to the Java language. But it is not a platform independent as the Java. So, its application runs only to the windows platform.

The main objective of this framework is to develop an application that can run on the windows platform. The current version of the .Net framework is 4.8.

Note: The .NET Framework is not only a language, but it is also a software and language neutral platform.



Components of .NET Framework

There are following components of .NET Framework:

1. CLR (Common Language Runtime)
2. CTS (Common Type System)
3. BCL (Base Class Library)
4. CLS (Common Language Specification)
5. FCL (Framework Class Library)
6. .NET Assemblies
7. XML Web Services
8. Window Services

CLR (common language runtime)

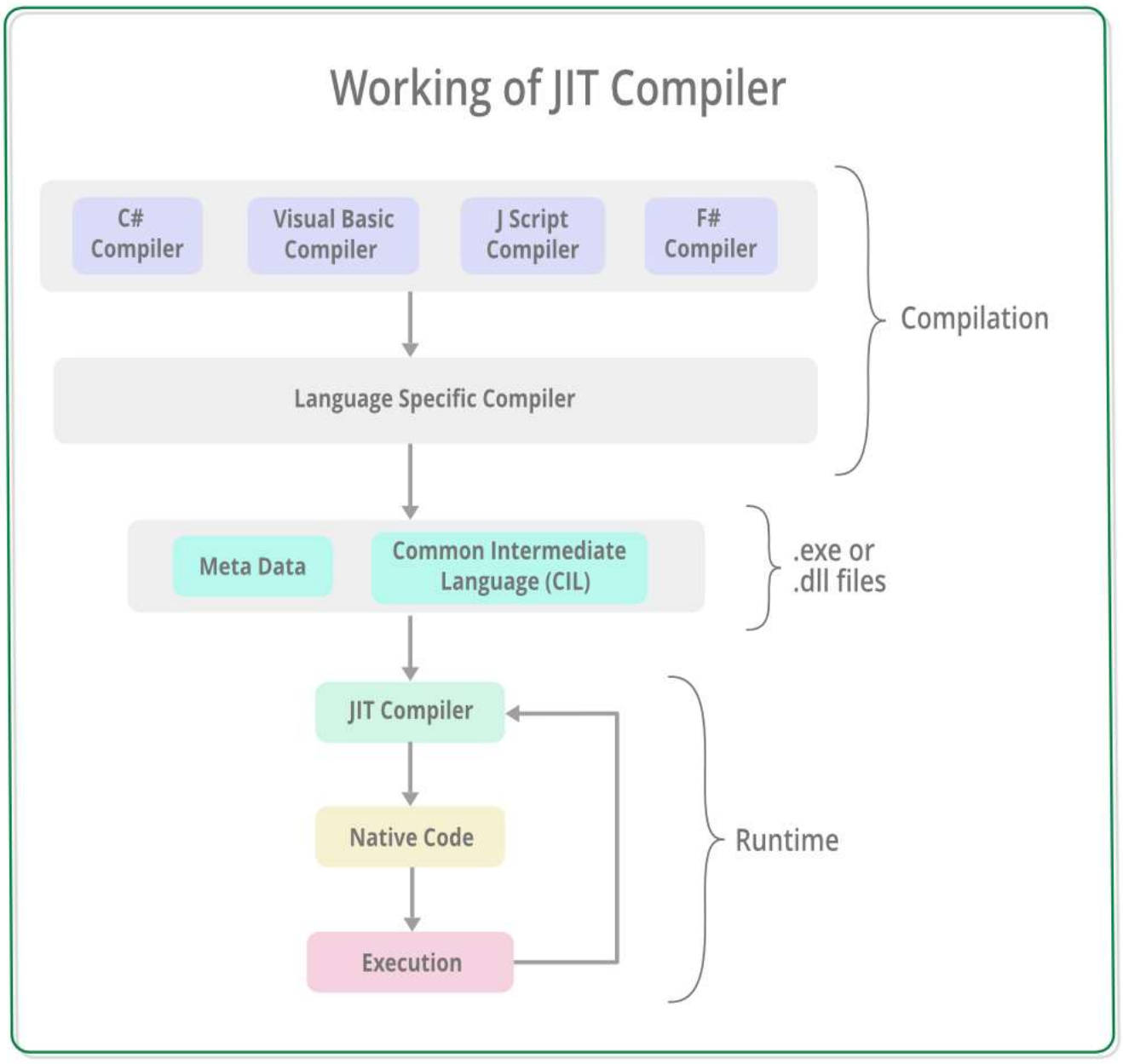
It is an important part of a .NET framework that works like a virtual component of the .NET Framework to executes the different languages program like C#, Visual Basic, etc. A CLR also helps to convert a source code into the byte code, and this byte code is known as CIL (Common Intermediate Language) or MSIL (Microsoft Intermediate Language). After converting into a byte code, a CLR uses a JIT compiler at run time that helps to convert a CIL or MSIL code into the machine or native code.

What is Just-In-Time(JIT) Compiler in .NET

Just-In-Time compiler(JIT) is a part of Common Language Runtime (CLR) in *.NET* which is responsible for managing the execution of *.NET* programs regardless of any *.NET* programming language. A language-specific compiler converts the source code to the intermediate language. This intermediate language is then converted into the machine code by the Just-In-Time (JIT) compiler. This machine code is specific to the computer environment that the JIT compiler runs on.

Working of JIT Compiler:

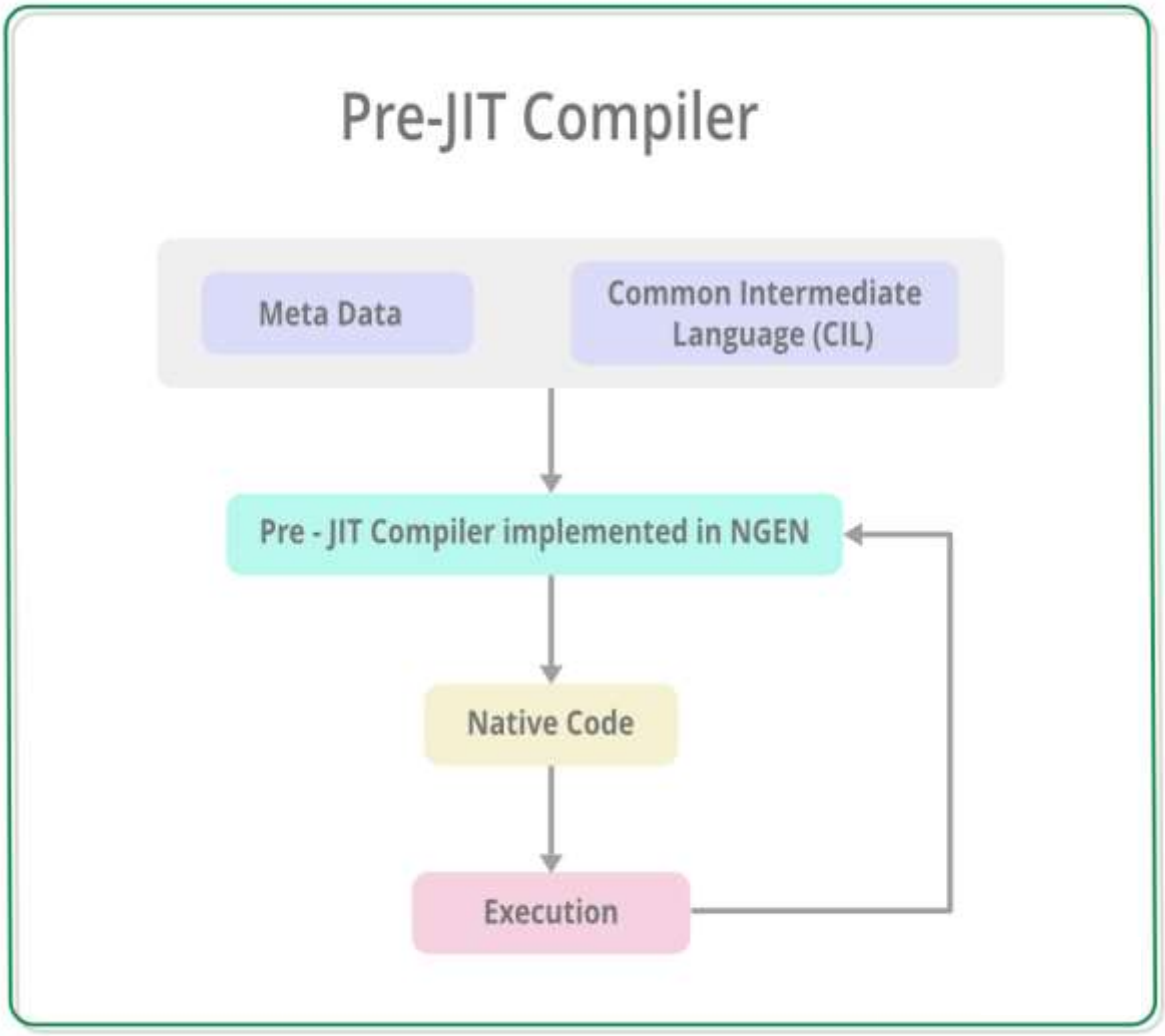
The JIT compiler is required to speed up the code execution and provide support for multiple platforms. Its working is given as follows:



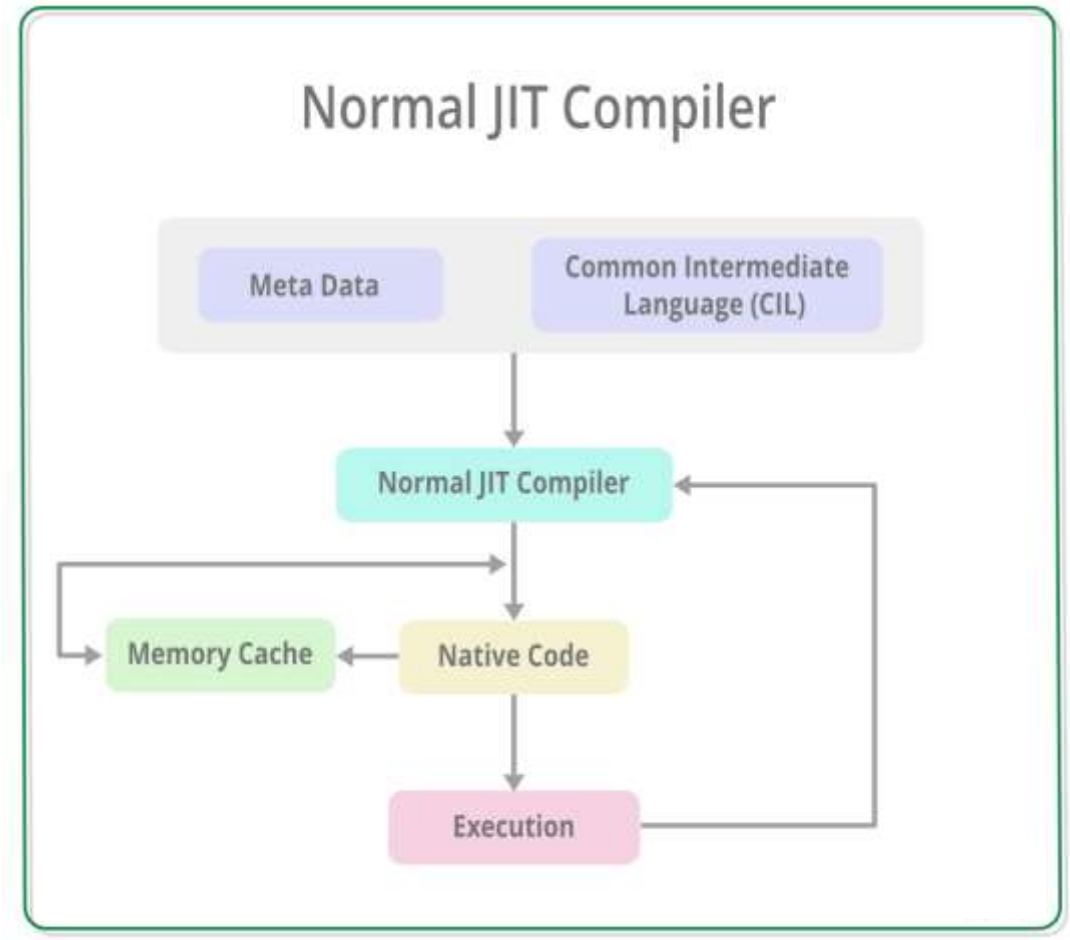
The JIT compiler converts the Microsoft Intermediate Language (MSIL) or Common Intermediate Language(CIL) into the machine code. This is done before the MSIL or CIL can be executed. The MSIL is converted into machine code on a requirement basis i.e. the JIT compiler compiles the MSIL or CIL as required rather than the whole of it. The compiled MSIL or CIL is stored so that it is available for subsequent calls if required.

Types of Just-In-Time Compiler: There are **three** types of JIT compilers which are as follows:

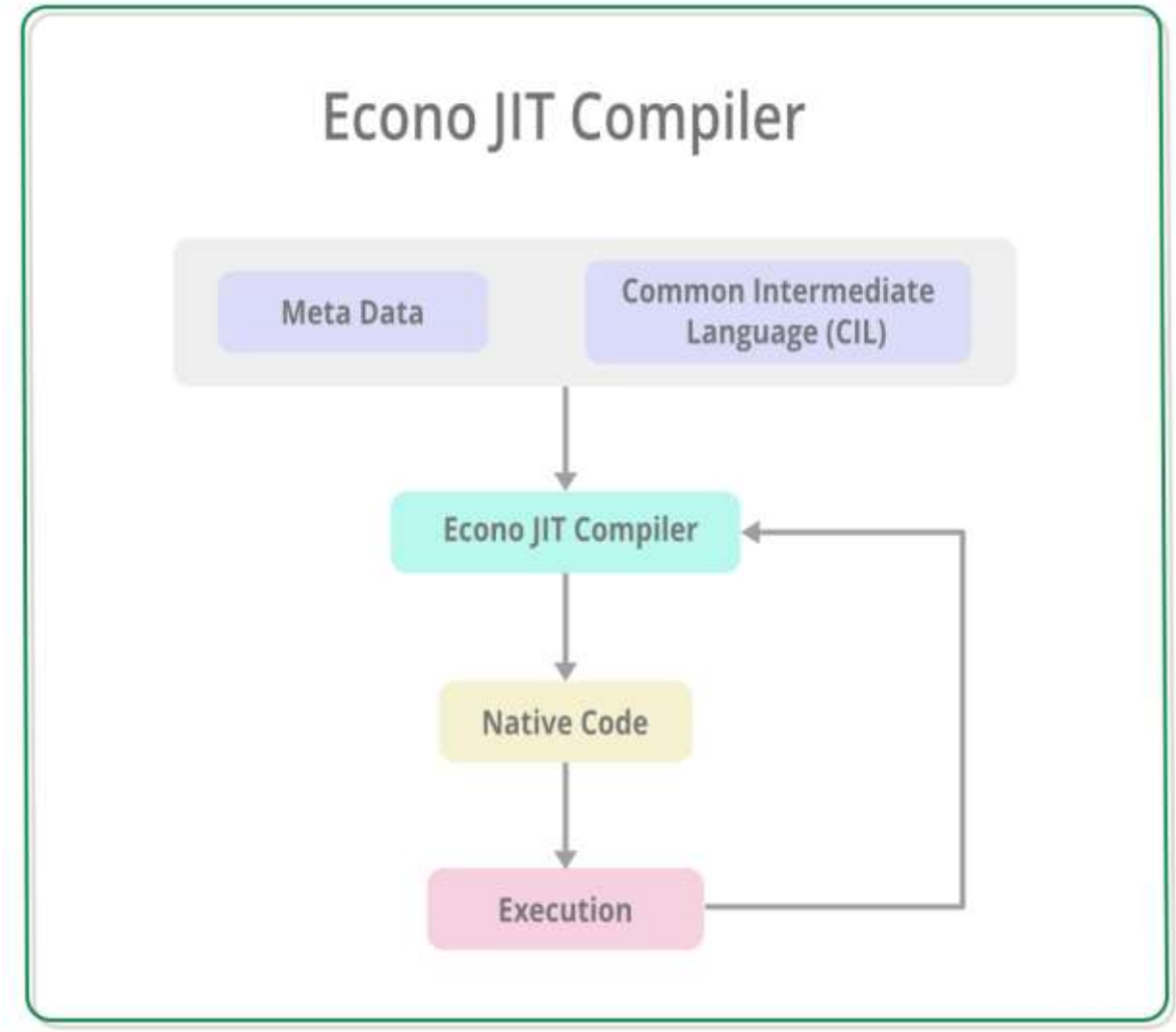
1. **Pre-JIT Compiler:** All the source code is compiled into the machine code at the same time in a single compilation cycle using the Pre-JIT Compiler. This compilation process is performed at application deployment time. And this compiler is always implemented in the Ngen.exe (Native Image Generator).



2. **Normal JIT Compiler:** The source code methods that are required at run-time are compiled into machine code the first time they are called by the Normal JIT Compiler. After that, they are stored in the cache and used whenever they are called again.



3. **Econo JIT Compiler:** The source code methods that are required at run-time are compiled into machine code by the Econo JIT Compiler. After these methods are not required anymore, they are removed.



CTS (Common Type System)

It specifies a standard that represent what type of data and value can be defined and managed in computer memory at runtime. A CTS ensures that programming data defined in various languages should beinteract with each other to share information. For example, in C# we define data type as int, while in VB.NET we define integer as a data type.

BCL (Base Class Library)

The base class library has a rich collection of libraries features and functions that help to implement many programming languages in the .NET Framework, such as C #, F #, Visual C ++, and more. Furthermore, BCL divides into two parts:

1. **User defined class library**
    - **Assemblies -** It is the collection of small parts of deployment an application's part. It contains either the DLL (Dynamic Link Library) or exe (Executable) file.
        1. In LL, it uses code reusability, whereas in exe it contains only output file/ or application.
        2. DLL file can't be open, whereas exe file can be open.
        3. DLL file can't be run individually, whereas in exe, it can run individually.
        4. In DLL file, there is no main method, whereas exe file has main method.
2. **Predefined class library**
    - **Namespace -** It is the collection of predefined class and method that present in .Net. In other languages such as, C we used header files, in java we used package similarly we used "using system" in .NET, where using is a keyword and system is a namespace.

CLS (Common language Specification)

It is a subset of common type system (CTS) that defines a set of rules and regulations which should be followed by every language that comes under the .net framework. In other words, a CLS language should be cross-language integration or interoperability. For example, in C# and VB.NET language, the C# language terminate each statement with semicolon, whereas in VB.NET it is not end with semicolon, and when these statements execute in .NET Framework, it provides a common platform to interact and share information with each other.

Microsoft .NET Assemblies

A .NET assembly is the main building block of the .NET Framework. It is a small unit of code that contains a logical compiled code in the Common Language infrastructure (CLI), which is used for deployment, security and versioning. It defines in two parts (process) DLL and library (exe) assemblies. When the .NET program is compiled, it generates a metadata with Microsoft Intermediate Language, which is stored in a file called Assembly.

FCL (Framework Class Library)

It provides the various system functionality in the .NET Framework, that includes classes, interfaces and data types, etc. to create multiple functions and different types of application such as desktop, web, mobile application, etc. In other words, it can be defined as, it provides a base on which various applications, controls and components are built in .NET Framework.

Key Components of FCL

1. Object type
2. Implementation of data structure
3. Base data types
4. Garbage collection
5. Security and database connectivity
6. Creating common platform for window and web-based application

Characteristics of .NET Framework

1. CLR (Common Language Runtime)
2. Namespace - Predefined class and function
3. Metadata and Assemblies
4. Application domains
5. It helps to configure and deploy the .net application
6. It provides form and web-based services
7. LINQ (Language Integrated Query - LINQ belongs to **System.Linq** namespace. LINQ allows you to write structured and type safe queries. It allows you to query local data as well as remote data.)
8. Security and Portability
9. Interoperability
10. It provides multiple environments for developing an application

Versions of .NET Framework

1. On 13 February 2002, Microsoft launched first version of .Net framework 1.0.
2. The second version 2.0 of .net framework was launched on 22 January 2006.
3. Third version 3.0 of .Net framework was released on 21 November 2006.
4. A .Net framework version 3.5 was released on 19 November 2007.
5. Version 4.0 of .Net framework was released on 29 September 2008
6. Version 4.5 of .Net framework was released on 15 August 2012.
7. .Net framework 4.5.1 version was announced on 17 October 2013
8. On 5 May 2014, a 4.5.2 version of .Net framework was released.
9. .Net framework 4.6 version was announced on 12 November 2014
10. .Net framework 4.6.1 version was released on 30 October 2015
11. .Net framework 4.6.2 version was announced on March 30, 2016
12. .Net framework 4.7 version was announced on April 5, 2017
13. .Net framework 4.7.1 version was announced on October 17, 2017
14. Version 4.7.2 of .Net framework was released on 30 April 2018.
15. And currently we are using .Net framework version 4.8 that was released on 18 April 2019