

Unit 1.1

Language Preliminaries & .Net Basic (8-Hrs)

Introduction to .Net

❖ Introduction

- The **framework** is a collection of software libraries. That are intended to perform specific tasks. It is used by programmers to build applications and websites.
- In other words, frameworks are a set of tools that are pre-build to create software.
- Creating software can be done in two ways. One is to write the codes from scratch and the other is by using a framework.
- **.NET** is a software framework which is designed and developed by Microsoft.
- One on side, it is a **developer platform** made up of tools, programming languages, and libraries for creating and building many different types of applications by integrating different programming language. On other side, .NET is an essential component of the operating system, that **helps in executing applications** for **general User**.
- It is used to develop **Window (Form)-based applications, Web-based applications, Console Application** and Web services.
- There is a variety of programming languages available on the .Net platform, VB.Net and C# being the most common ones..NET supports more than 60 programming languages in which 11 programming languages are designed and developed by Microsoft.
- There are various implementations of .NET Standard
 - **.NET Framework** is the original implementation of .NET. It supports running websites, services, desktop apps, and more on Windows. .NET Framework is a **Windows-only version** of .NET for building any type of app that runs on **Windows**.
 - **.NET Core** is a **cross-platform** implementation for running websites, services, desktop apps and console apps on Windows, Linux, and macOS. .NET Core is a cross-platform version of .NET for building websites, services, and console apps. It is open source on GitHub and accepts contribution from community.
 - **Xamarin/Mono** is a .NET implementation for running apps on all the major mobile operating systems, including iOS and Android. Xamarin extends the .NET platform with tools and libraries specifically for building apps for Android, iOS, and Windows. So using xamarin, we can create native android, IOS and windows apps using existing .net Skills.

❖ Architecture of .NET Framework

- The two major components of .NET Framework are the

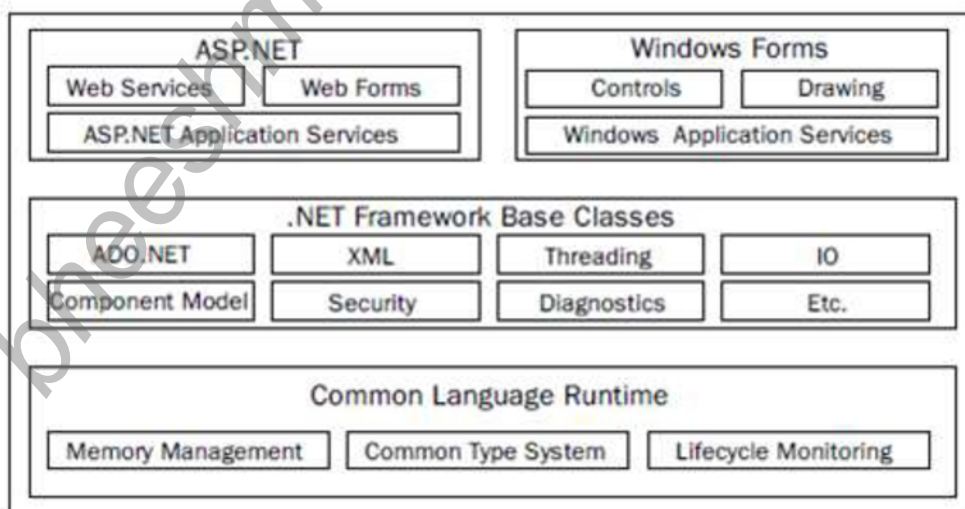


Fig: Basic Components of .NET Framework

- i. Common Language Runtime(CLR)
 - .Net Framework provides runtime environment called Common Language Runtime (CLR) for all .NET program. The Common Language Runtime (CLR) is the execution engine that handles running applications.

- CLR manages and executes code written in any .net language like c#, VB etc.
 - CLR is an agent that **manages code** at execution time, providing core services such as memory management & garbage collection (manage allocation and release of memory for an application), exception handling (managing runtime error), thread management (Multithreading and thread management) while also enforcing strict type safety (i.e. `int a=5.5` is not allowed until we force explicit type conversion) and other forms of code accuracy. **The Common Type System (CTS)** is a standard that specifies how Type definitions and specific values of Types are represented in computer memory. It is intended to allow programs written in different programming languages to easily share information. For example, an integer variable in C# is written as `int`, whereas in Visual Basic it is written as `integer`. It defines how to declare, use and manage type in CLR and **support cross language integration** (code written in one language can interact with the code written in another language that provides Language independence). Therefore, in .Net Framework we have single class called `System.Int32` to interpret these variables
 - The code which runs under the CLR is also called as **Managed Code**.
- ii. .NET Framework Class Library.
- It is huge library of classes and interfaces available to all languages using .NET framework. The Class Library provides a set of APIs (Application Programming Interface) and types for common functionality that help in **speeding up the development process**.
 - It provides types for strings, dates, numbers, etc.
 - The Class Library includes APIs for reading and writing files, graphics drawing and rendering, connecting to databases, and more.
 - .NET Framework consists of classes, interfaces and value types that help in speeding up the development process and provide access to system functionality.
 - It is made up of a hierarchy of **namespaces**. **Namespaces** providing logical grouping of several classes.
- Example:
- i. **System** namespace contains functionality available for accessing **Console** Class.
 - ii. **System.Data** namespace contains functionality available for accessing databases. **System.Data.SqlClient** provides functionality specific to SQL server.

❖ Compilation and Execution of .NET application

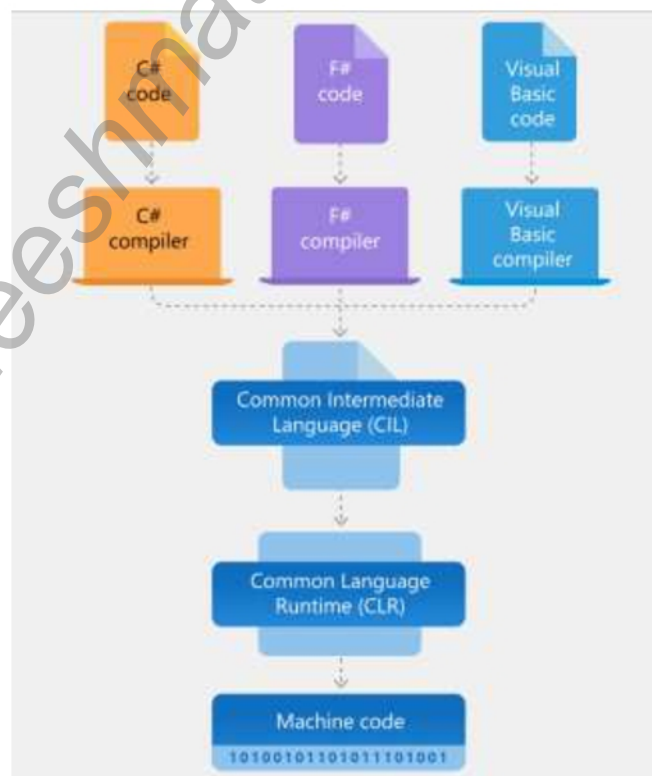


Fig: Compilation and Execution of .NET application

- .NET applications are written in .NET compatible languages like C#, F#, or Visual Basic programming language.
- Code is compiled into a language-agnostic **Common Intermediate Language (CIL)**. So .NET compatible language is **compiled** to platform neutral language called CIL. Compiled code is stored in **assemblies**—files with a .dll or .exe file extension. [Similar to Bytecode in Java].
- MSIL or IL (**Microsoft intermediate language (MSIL)** or **Intermediate language (IL)**) is independent of programming language, targeting machine and OS i.e. Platform Independent code.
- When the source code is compiled into IL, the required metadata is generated.
- The metadata contains the definition of types, member signature, the members in the code and other details that the code uses at the time of execution.
- The **CLR (Common Language runtime)** locates and extracts the metadata from the file during execution.
- For execution, the platform specific common language runtime (CLR) takes the CIL i.e. assembly and uses a just-in-time compiler (JIT) [**Interpreter**] to turn/interpret it into machine specific code/Native Code that can execute on the **specific architecture** of the computer it is running on i.e. Converts MSIL to native code.
- But soon after compiling, the code is passed through verification process that examines the IL and metadata to check whether the code is **type safe or not** i.e. **Type safe code access only those memory locations which it is authorized to access. Example, type-safe code cannot read values from another object's private fields**

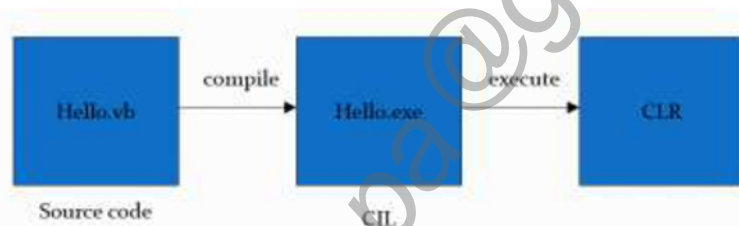


Fig: CIL operation

❖ Benefits of .Net Framework

1. Consistent Programming Model

- Net Framework provides a consistent object oriented programming model to create program across different languages. This model is used to create programs for performing different tasks like connecting to and retrieving data from databases and reading from and writing to files etc.

2. Cross Platform support

- Any Windows platform that supports CLR can execute .Net application.
- .Net application enables interoperability between different multiple windows operating system

3. Language interoperability

- Language interoperability enables any compiled code written in any .NET language to be reused from other .NET language and interact with each other. This allows reusability of code and improves the efficiency of the development process.
- Example: Visual basic class can be inherited in c# and vice versa.
- CLR has built in support for Language interoperability (Common Type System).

4. Automatic Management of Resources

- .NET Framework provides a feature called CLR that automatically tracks the resource (files, memory, database connection) usage and relieves the users from the tasks of manual resource management.
- We don't need to manually free the application resources, such as files, memory, network and Database connection

5. Ease of Deployment

- .NET Framework makes the task of deployment easier. In .NET, applications are deployed in the form of assemblies [.exe or DLL (Dynamic Link Library)].
- Assemblies also store information about different versions of a single component used by an application.

❖ .NET Application

- Major .NET applications are
 1. **Window Based:** Example, Desktop Based Application
 2. **Web Based Application:** Example, Asp.NET application

3. **Console Application:**

A console application is a computer program designed to be used via a text-only computer interface, such as a text terminal, or the command line interface (CLI) or text-based interface.

4. **Mobile Application:** .NET and C# can be used to create native apps for the billions of Android, iPhone, iPad, Mac, and Windows devices around the world.

5. **Web Service:**

- Web service are the standardized way for developing interoperable application i.e. enabling and application to invoke a method of another application.
- Web service can be on the same or remote computer.
- HTTP protocol is widely used by web service to send and receive message.
- Example: A java application can interoperate with a web service build using .net and viceversa.

Assignment-1:

1. Write short notes on
 - a. CLS (Common Language Specification)
 - b. .NET for Developer vs .NET for User
 - c. .NET Framework vs .NET Core
 - d. Dynamic Link Library(DLL)