Charutar Vidya Mandal's
SEMCOM
Vallabh Vidyanagar

Faculty Name: Mikita Bhatiya
Subject: Desktop Application **Class: SYBCA**
Unit – 1: Introduction to .NET Framework & VB.NET

TOPICS

- .NET Architecture, .NET Languages;
- Microsoft Intermediate Language (MSIL),
- The Just-In-Time (JIT) Compiler;
- VB.NET – Introduction, Applications and Types of Project
- Introduction to Visual Studio IDE
- Creating simple Windows Application using VB.NET
- Variables, Data Types, Constants, Operators

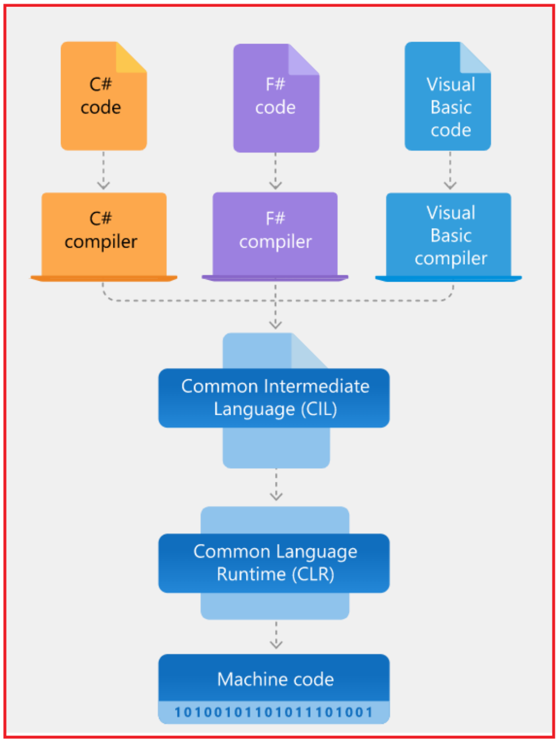
➢ Introduction to .NET Framework:

- ✓ NET stands for Network Enabled Technology (Internet). In .NET, dot (.) refers to Object-Oriented, and NET refers to the internet. So, the complete .NET means through Object-Oriented we can implement internet-based applications.
- ✓ According to Microsoft, .NET is a Free, Cross-Platform, Open-Source developer platform for building many different types of applications.
- ✓ With .NET, we can use multiple languages (C#, VB, F#, etc.), Editors (Visual Studio, Visual Studio Code, Visual Studio for Mac, etc.), and Libraries to build applications for Web, Mobile, Desktop, Games, and more.
- ✓ A framework is software. It is a collection of many small technologies integrated together to develop applications that can be executed anywhere.
- ✓ .NET Framework is a software development platform developed by Microsoft for building and running Windows applications.
- ✓ The .NET framework consists of developer tools, programming languages, and libraries to build desktop and web applications.
- ✓ It is also used to build websites, web services, and games.
- ✓ The .NET framework was meant to create applications, which would run on the Windows Platform.
- ✓ The first version of the .NET framework was released in the year 2002. The version was called .NET framework 1.0. The Microsoft .NET framework has come a long way since then, and the current version is .NET Framework 4.8.1.
- ✓ The Microsoft .NET framework can be used to create both – Form-based and Web-based applications.

- ✓ The framework also supports various programming languages such as Visual Basic and C#. So developers can choose and select the language to develop the required application.

➢ .NET Framework Architecture

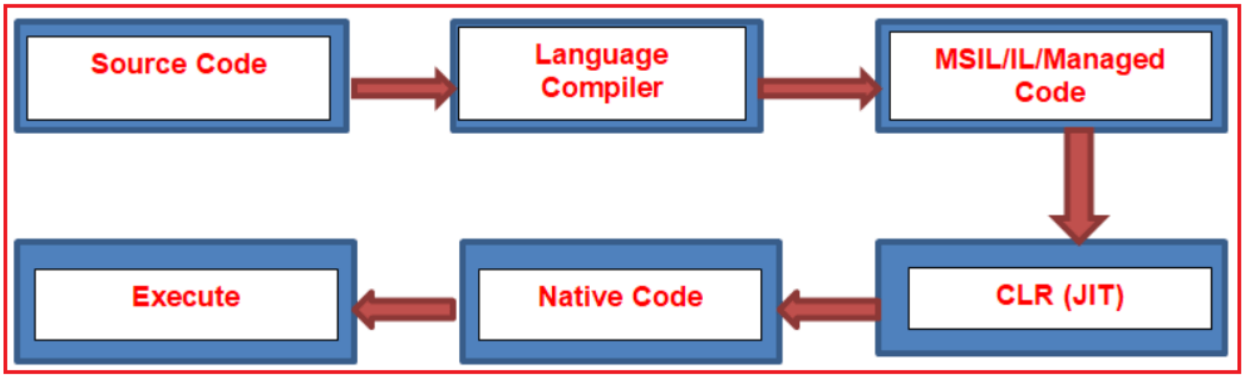
- ✓ .NET Framework Architecture is a programming model for the .NET platform that provides an execution environment and integration with various programming languages for simple development and deployment of various Windows and desktop applications.
- ✓ It is also used to create a form based, mobile and web-based application or services that are available in Microsoft environment. Furthermore, the .NET framework is a pure object oriented, similar to the Java language. But it is not platform independent as Java. So, its application runs only on the windows platform.
- ✓ It consists of class libraries and reusable components.
- ✓ It includes a large library of pre-coded solutions to common programming problems.
- ✓ A class library is a collection of methods and functions that can be used for the purpose of application development.
- ✓ The class library is used by programmers, who combine it with their own code to produce applications.
- ✓ The main objective of this framework is to develop an application that can run on the windows platform.
- ✓ The two major components of the .NET Framework are the Common Language Runtime and the .NET Framework Class Library.



➢ Components of the .NET Framework

<u>Common Language Runtime (CLR)</u>:.NET provides a run-time environment called the common language runtime that runs the code and provides services that make the development process easier.

- ✓ It handles the execution of code and provides useful services for the implementation of the program.
- ✓ In .NET, programs are not compiled into executable files; they are compiled into Microsoft Intermediate Language (MSIL) files (also known as IL or CIL), which the CLR then executes. The MSIL files that VB.NET produces are identical to the IL files that other .NET languages produce. A key fact about the CLR is that it is common; the same runtime supports development in C# as well as in VB.NET.
- ✓ CLR is the core component under the .NET framework which is responsible for converting the MSIL (Microsoft Intermediate Language) code into native code.
- ✓ After converting into CIL, a CLR uses a JIT compiler at run time that helps to convert a CIL or MSIL code into the machine or native code.



- ✓ **In the .NET framework, the code is compiled twice.**
- ✓ In the 1st compilation, the source code is compiled by the respective language compiler and generates the intermediate code which is known as **MSIL (Microsoft Intermediate Language)** or **IL (Intermediate language code),** or **Managed Code**.
- ✓ In the 2nd compilation, **MSIL** is converted into **Native code** (native code means code specific to the Operating system so that the code is executed by the Operating System).

<u>Framework/Base Class Library (FCL/BCL):</u> Base Class Libraries (BCL) are designed by Microsoft. Without BCL we can't write any code in .NET. So, BCL is also known as the basic building block of .NET Programs. BCL contains pre-defined classes and these classes are used for the purpose of application development.

- ✓ A class library is a collection of methods and functions that can be used for the purpose of application development.
- ✓ For example, there is a class library with methods to handle all file-level operations. So there is a method which can be used to read the text from a file. Similarly, there is a method to write text to a file.

➢ .NET Languages

- ✓ **.NET Languages** are computer programming languages that are used to produce programs that execute within the Microsoft .NET Framework.
- ✓ Regardless of which .NET language is used, the output of the language compiler is a representation of the same logic in an intermediate language named Common Intermediate Language.
- ✓ While there are currently over 40 languages with compilers for the .NET Framework, only a small number of them are widely used and supported by Microsoft. The remainder is composed of languages developed by third party vendors (non-Microsoft .NET languages).
- ✓ **Microsoft .NET languages:** C#, Visual Basic .NET, C++, J#, JScript.NET, Windows PowerShell, IronPython, IronRuby & F#.
- ✓ **Third-party languages:** Microsoft encourages third party vendors to make use of Visual Studio. Net. Third, party vendors can write compilers for different languages that compile the language to MSIL. They can easily use Visual Studio.NET as an IDE for their .NET compliant language. Several third-party languages are supporting the .NET platform. These languages include COBOL, Pascal, Perl, Python, Smalltalk, etc.

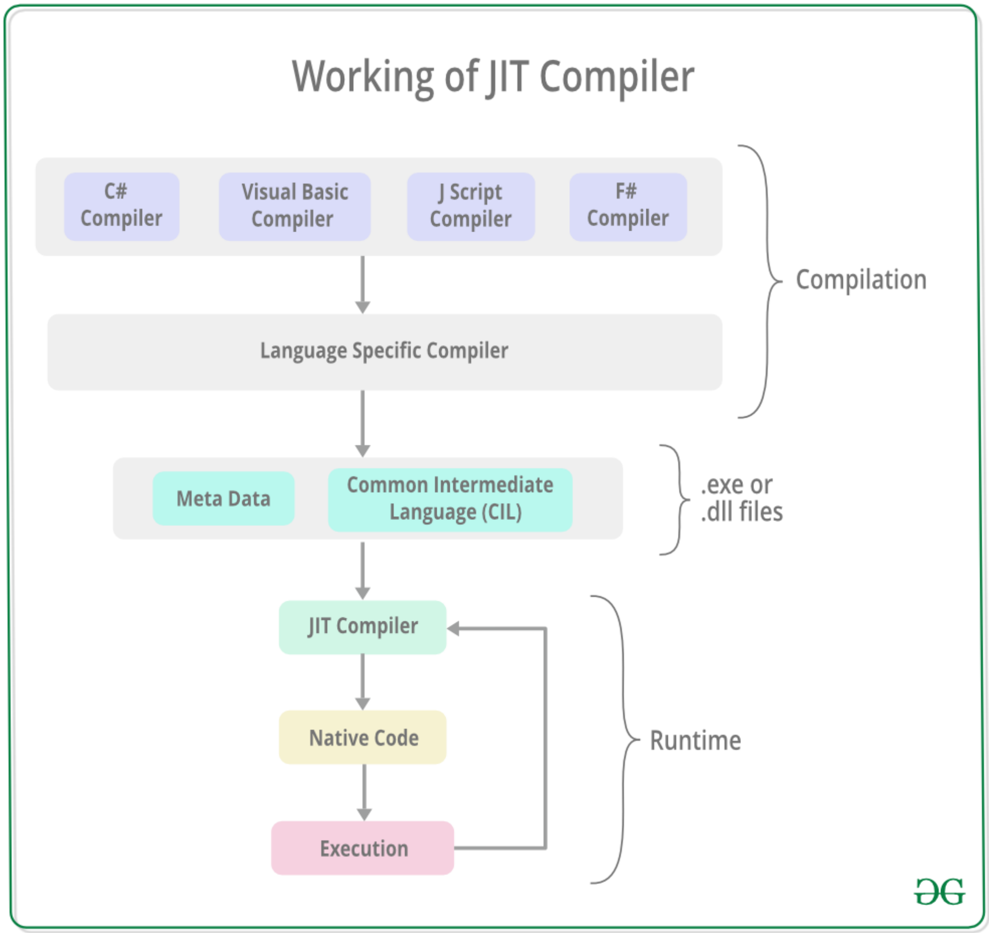
➢ Microsoft Intermediate Language (MSIL)

- A .NET programming language (C#, VB.NET, J# etc.) does not compile into executable code; instead, it compiles into an intermediate code called Microsoft Intermediate Language (MSIL). As a programmer one need not worry about the syntax of MSIL - since our source code is automatically converted to MSIL. The MSIL code is then send to the CLR (Common Language Runtime) that converts the code to machine language which is then run on the host machine.
- All .NET languages compile code to this IL, which is known as Microsoft Intermediate Language, MSIL, or CIL.
- MSIL is a common language in the sense that the same programming tasks written with different .NET languages produce the same IL code.
- When a compiler produces Microsoft Intermediate Language (MSIL), it also produces Metadata (a set of data that describes and gives information about other data).
- Microsoft Intermediate Language (MSIL) includes instructions for loading, storing, initializing, and calling methods on objects, as well as instructions for arithmetic and logical operations, control flow, direct memory access, exception handling, and other operations.
- **Advantages**

- ✓ It offers cross− language integration, including cross− language inheritance, which allows you to create a new class by deriving it from a base class written in another language.
- ✓ It facilitates automatic memory management, known as garbage collection.
- ✓ Compilation is much quicker
- ✓ It allows you to compile code once and then run it on any CPU and operating system that supports the runtime.

➢ Just-In-Time Compiler

- ✓ 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:



➢ VB.NET Introduction

- ✓ VB.NET stands for Visual Basic.NET, and it is a computer programming language developed by Microsoft.
- ✓ It was first released in 2002 to replace Visual Basic 6.
- ✓ VB.NET is an object-oriented programming language. This means that it supports the features of object-oriented programming which include encapsulation, polymorphism, abstraction, and inheritance.
- ✓ Visual Basic.NET runs on the .NET framework, which means that it has full access to the .NET libraries.