Introduction

Course Outcome:

CO1 : Use various tools used in dynamic web page designing & hosting of Web sites
CO2 : Use GUI tools of .net framework.
CO3 : Use basic and advance .net controls.
CO4 : Build applications integrated with .net Framework.
CO5 : Apply Database Controls to establish database connectivity.

dot Net?

.Net is an open source developer platform, created by Microsoft, for building many different types of applications.

.NET is a free, cross-platform, open source developer platform for building many different types of applications.

With .NET, you can use multiple languages, editors, and libraries to build for web, mobile, desktop, games, and IoT.

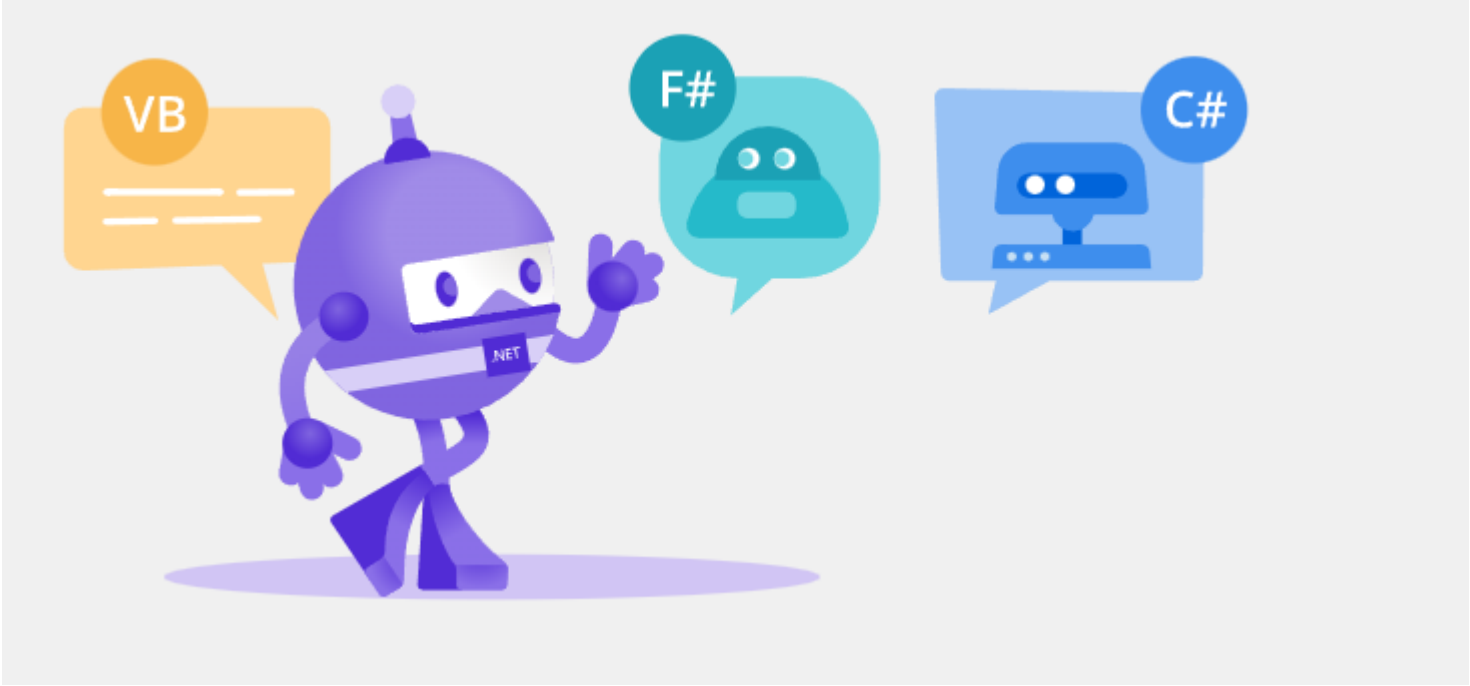
Languages

You can write .NET apps in C#, F#, or Visual Basic.

•**C#** is a simple, modern, object-oriented, and type-safe programming language.

•**F#** is a programming language that makes it easy to write succinct, robust, and performant code.

•**Visual Basic** is an approachable language with a simple syntax for building type-safe, object-oriented apps.



Cross Platform
Whether you're working in C#, F#, or Visual Basic, your code will run natively on any compatible OS. Different .NET implementations handle the heavy lifting for you:

•**.NET** is a cross-platform .NET implementation for websites, servers, and console apps on Windows, Linux, and macOS.

•**.NET Framework** supports websites, services, desktop apps, and more on Windows.

•**Xamarin/Mono** is a .NET implementation for running apps on all the major mobile operating systems.

One consistent API

.NET Standard is a base set of APIs that are common to all .NET implementations.

Libraries

To extend functionality, Microsoft and others maintain a healthy package ecosystem built on .NET Standard.
NuGet is a package manager built specifically for .NET that contains over 90,000 packages.

Active community and open-source

.NET is open source and under the .NET Foundation. The .NET Foundation is an independent organization to foster open development and collaboration around the .NET ecosystem.
Because .NET is open source, you can join the thousands of developers and companies already contributing to the .NET platform.

Tools

The Visual Studio product family provides a great .NET development experience on Windows, Linux, and macOS.
The Visual Studio Marketplace has thousands of editor extensions from Microsoft and others.

Why dot Net?

Productive

.NET helps you develop high quality applications faster. Modern language constructs like generics, Language Integrated Query (LINQ), and asynchronous programming make developers productive.
Combined with the extensive class libraries, common APIs, multi-language support, and the powerful tooling provided by the Visual Studio family, .NET is the most productive platform for developers.

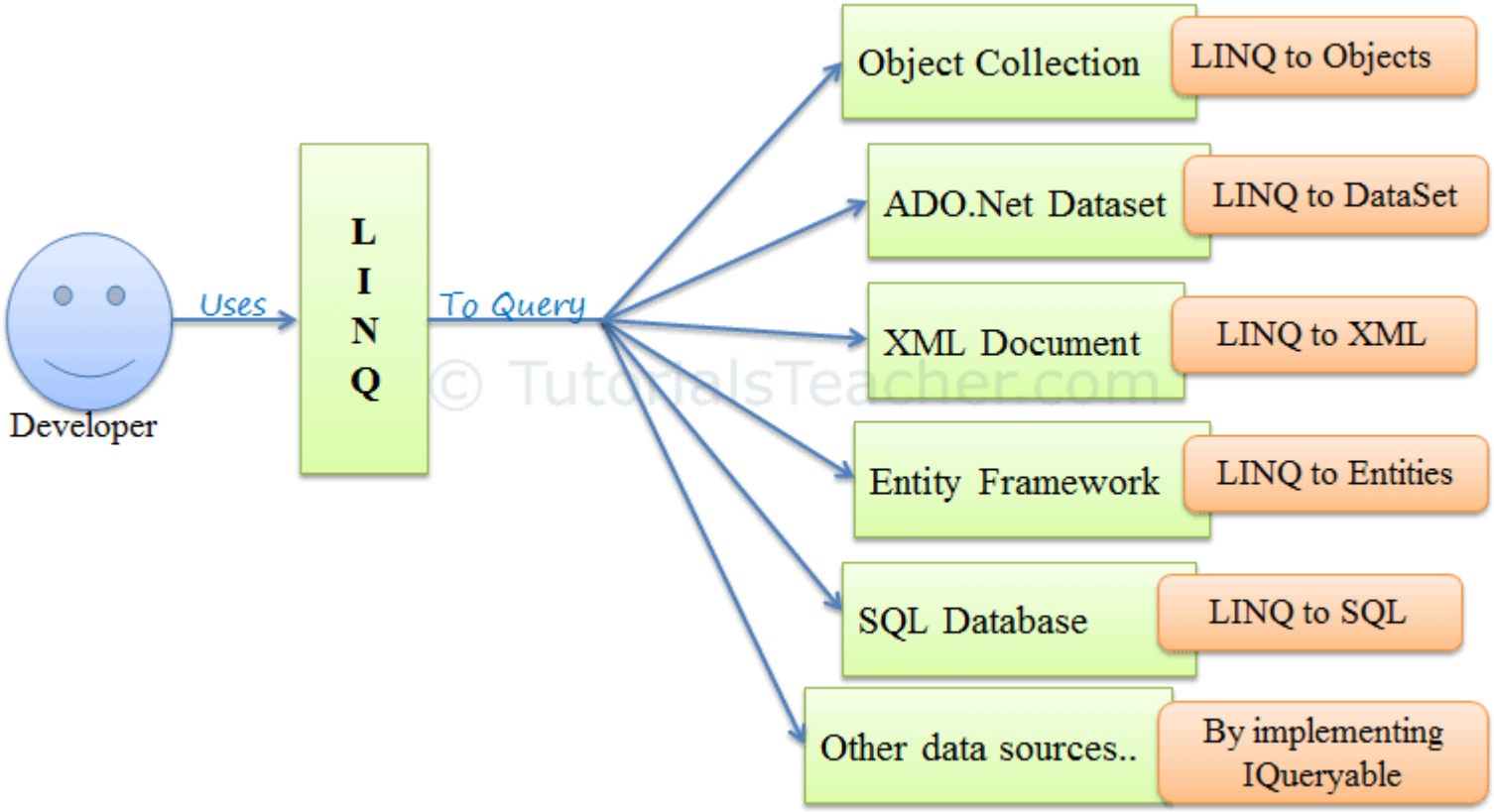
Any app, any platform

With .NET you can target any application type running on any platform. Developers can reuse skills and code across all of them in a familiar environment. That means developers can build apps faster, with less cost.
From mobile applications running on iOS, Android, and Windows, to Enterprise server applications running on Windows Server and Linux, or high-scale microservices running in the cloud, .NET provides a solution for you.

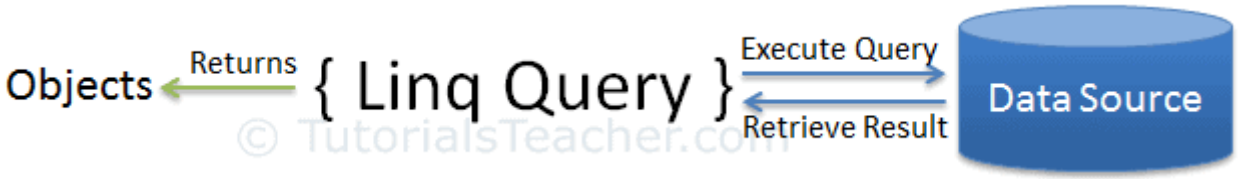
Loved by developers

.NET is a modern, innovative, open source development platform and developers love it. .NET ranked as the #1 most-loved framework on the Stack Overflow Developer Survey in both 2019 and 2020 editions.

- LINQ (Language Integrated Query) is uniform query syntax in C# and VB.NET to retrieve data from different sources and formats. It is integrated in C# or VB, thereby eliminating the mismatch between programming languages and databases, as well as providing a single querying interface for different types of data sources.
- For example, SQL is a Structured Query Language used to save and retrieve data from a database. In the same way, LINQ is a structured query syntax built in C# and VB.NET to retrieve data from different types of data sources such as collections, ADO.Net DataSet, XML Docs, web service and MS SQL Server and other databases.


Developer
Uses
L
I
N
Q
To Query
Object Collection
LINQ to Objects
ADO.Net Dataset
LINQ to DataSet
XML Document
LINQ to XML
Entity Framework
LINQ to Entities
SQL Database
LINQ to SQL
Other data sources..
By implementing IQueryable

LINQ Usage


Objects
Returns
{ Linq Query }
Execute Query
Retrieve Result
Data Source

Performance where it matters:

.NET is fast. Really fast! That means applications provide better response times and require less compute power.
The popular TechEmpower benchmark compares web application frameworks with tasks like JSON serialization, database access, and server side template rendering - .NET performs faster than any other popular framework.

Trusted and secure

The .NET platform is officially supported by Microsoft and trusted by thousands of companies and millions of developers. Microsoft takes security very seriously and releases updates quickly when threats are discovered.

Large ecosystem

With over 5,000,000 .NET developers worldwide, you can leverage the large ecosystem by incorporating libraries from the NuGet package manager and the Visual Studio Marketplace.

Open source

The .NET Foundation is an independent non-profit supporting the innovative, commercially friendly, open source .NET ecosystem. .NET has over 100,000 contributions from developers from over 3,700 companies outside of Microsoft.

Introduction to .NET Framework

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

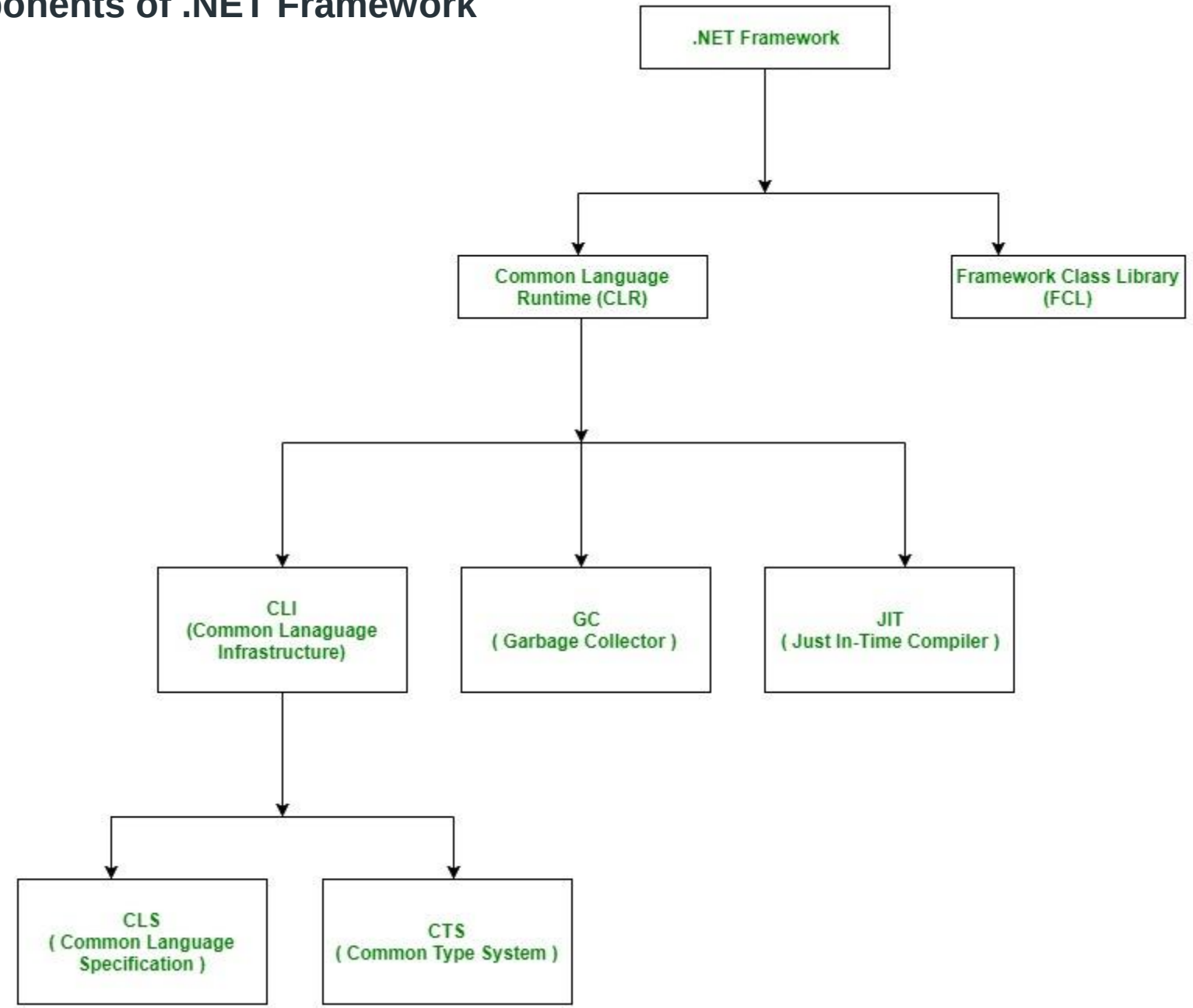
What is .NET Framework:

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.

.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.

Main Components of .NET Framework



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).

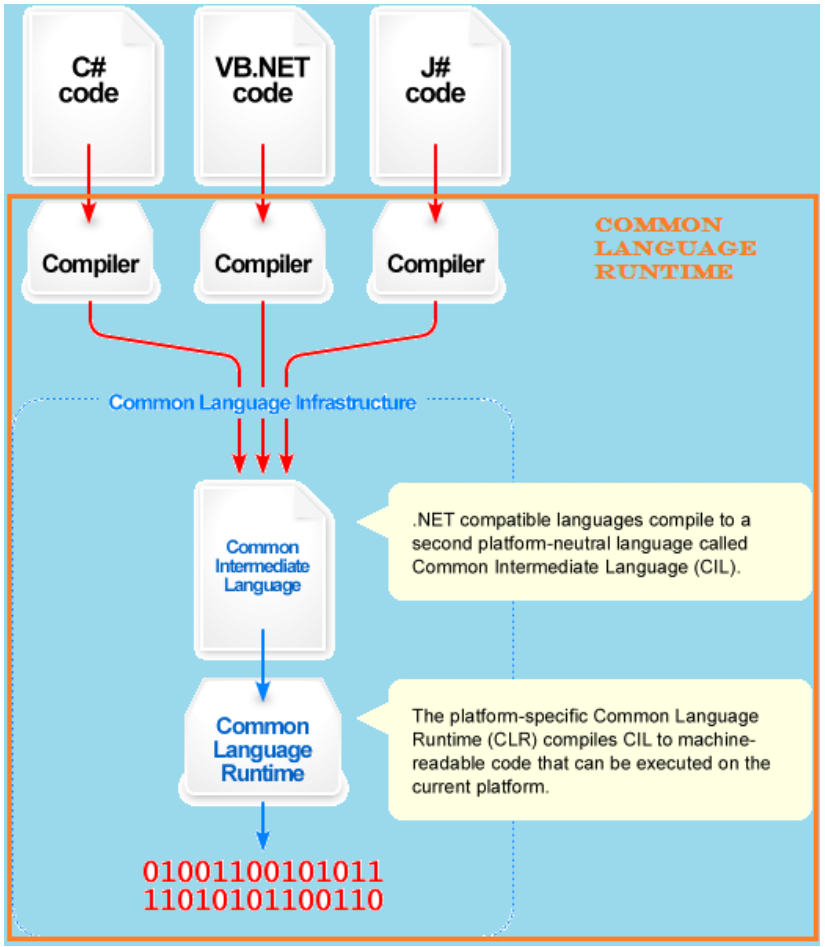
After converting into a byte code, a CLR uses a JIT compiler at run time that helps to convert a CIL or MSIL code into the machine or native code.

It is the run-time environment in the .NET Framework that runs the codes and helps in making the development process easier by providing various services such as remoting, thread management, type-safety, memory management, robustness, etc..

Basically, it is responsible for managing the execution of .NET programs regardless of any .NET programming language.

Common Language Infrastructure (CLI)

CLI is a specification developed by Microsoft that describes the executable code and runtime environment. In simple terms this allows us to use various high-level programming languages on various machines without rewriting the code.



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.

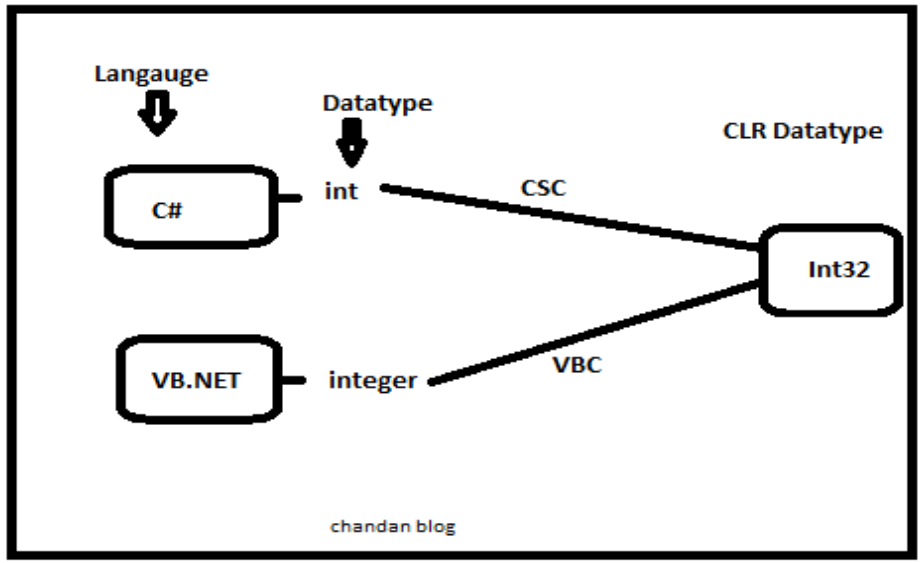
For example, in C# and VB.NET language, the C# language terminate each statement with semicolon, whereas in VB.NET it is not end with semicolon, and when these statements execute in .NET Framework, it provides a common platform to interact and share information with each other.

CTS

Common Type System (CTS) describes the datatypes that can be used by managed code. CTS defines how these types are declared, used and managed in the runtime. It facilitates cross-language integration, type safety, and high-performance code execution. The rules defined in CTS can be used to define your own classes and values.

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 be interact 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.



C# has an int data type and VB.NET has Integer data type. Hence a variable declared as an int in C# and Integer in VB.NET, finally after compilation, uses the same structure Int32 from CTS.

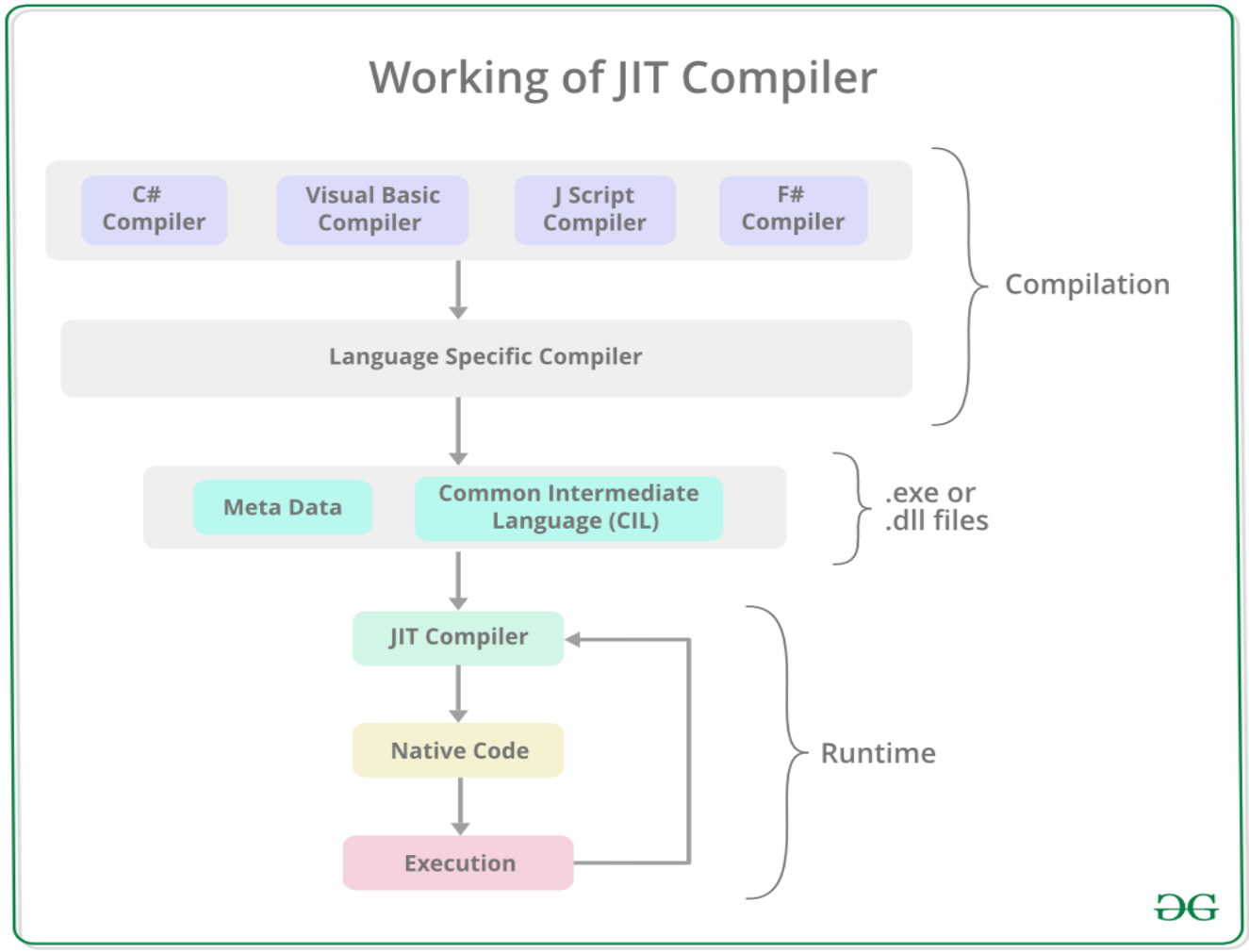
Garbage Collector

Garbage collection makes automatic memory management in .NET framework possible. Thanks to it, the developer is only responsible for allocating memory by creating new instances of objects. Allocated memory is automatically released by the garbage collector once the created objects are not used any more.

Just-In-Time(JIT)

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.

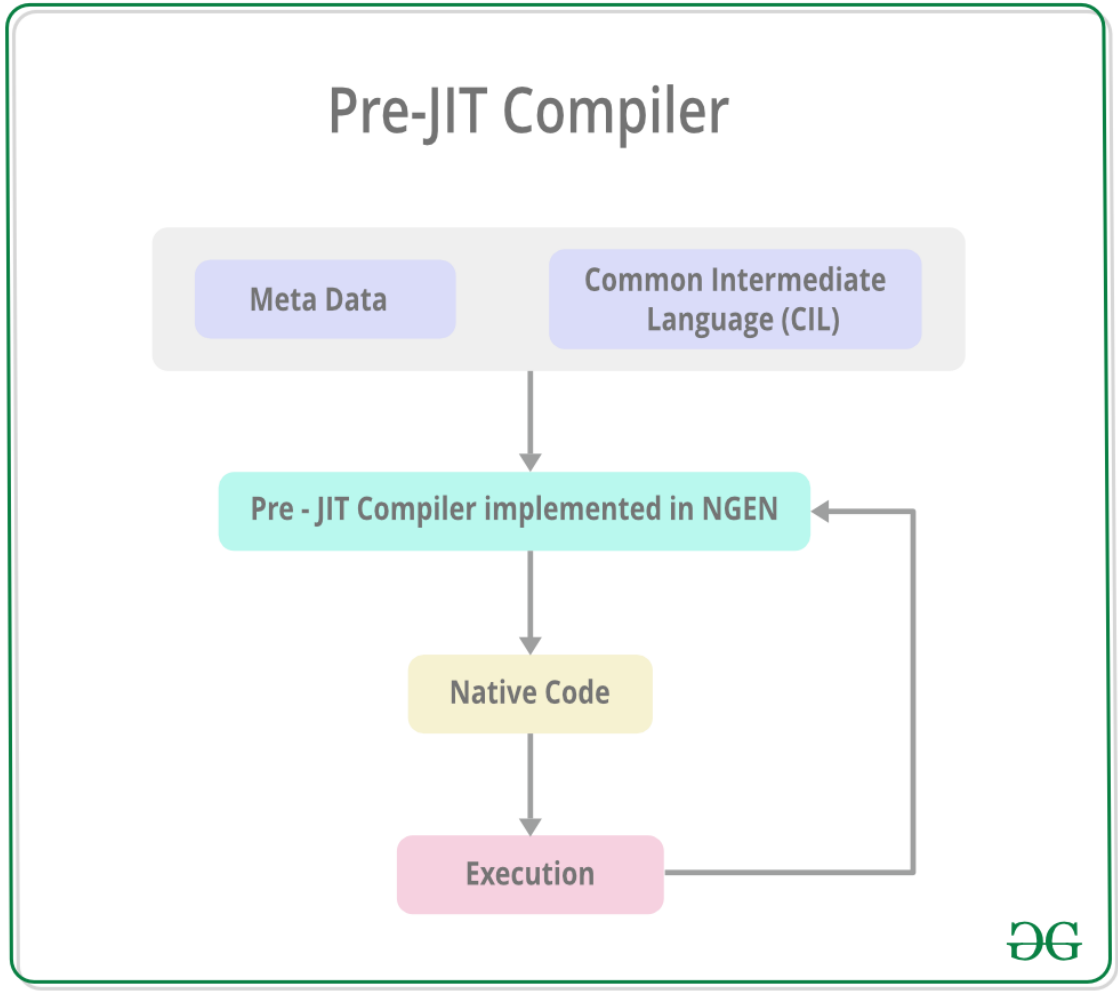
Working of JIT Compiler: The JIT compiler is required to speed up the code execution and provide support for multiple platforms. Its working is given as follows:



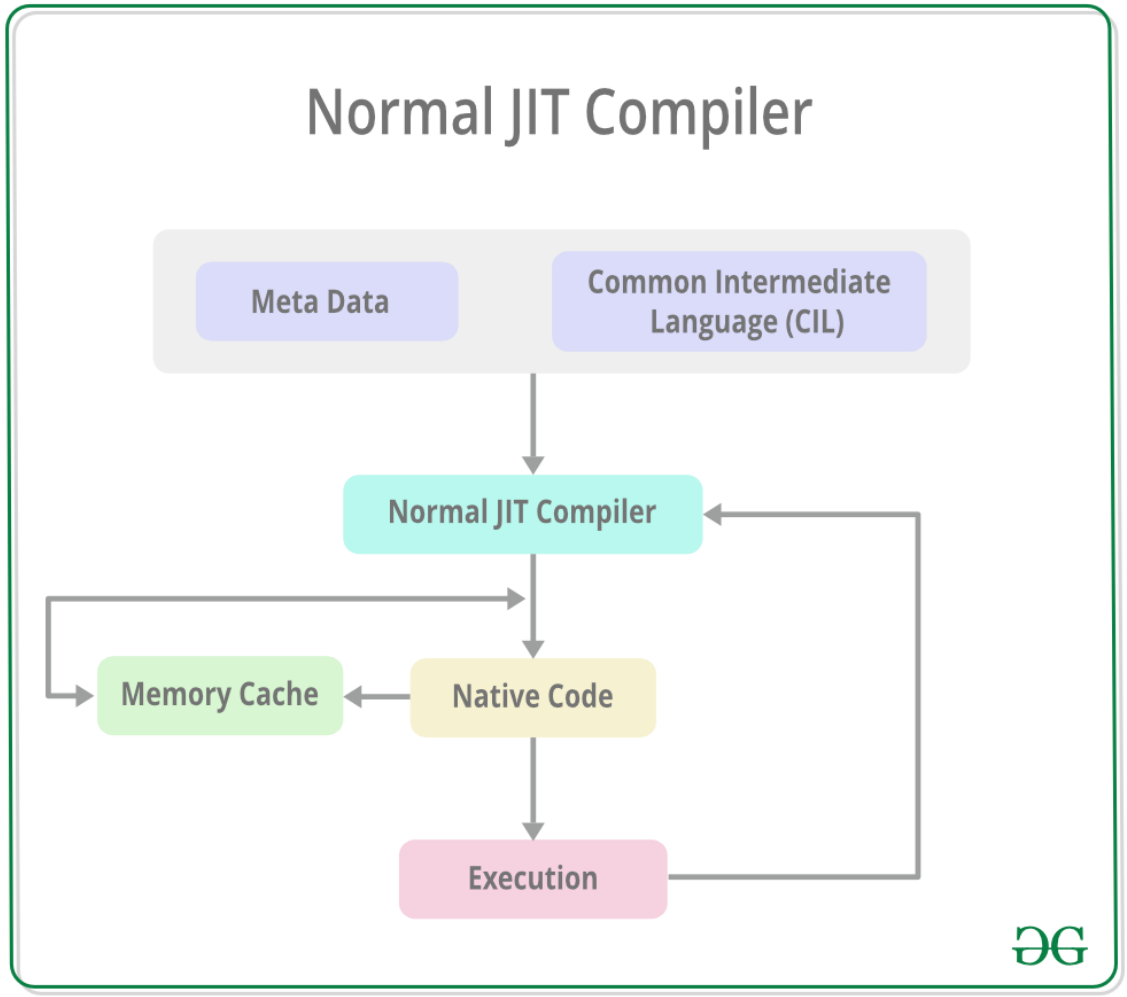
The JIT compiler converts the Microsoft Intermediate Language(MSIL) or Common Intermediate Language(CIL) into the machine code. This is done before the MSIL or CIL can be executed. The MSIL is converted into machine code on a requirement basis i.e. the JIT compiler compiles the MSIL or CIL as required 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 **3** types of JIT compilers which are as follows:

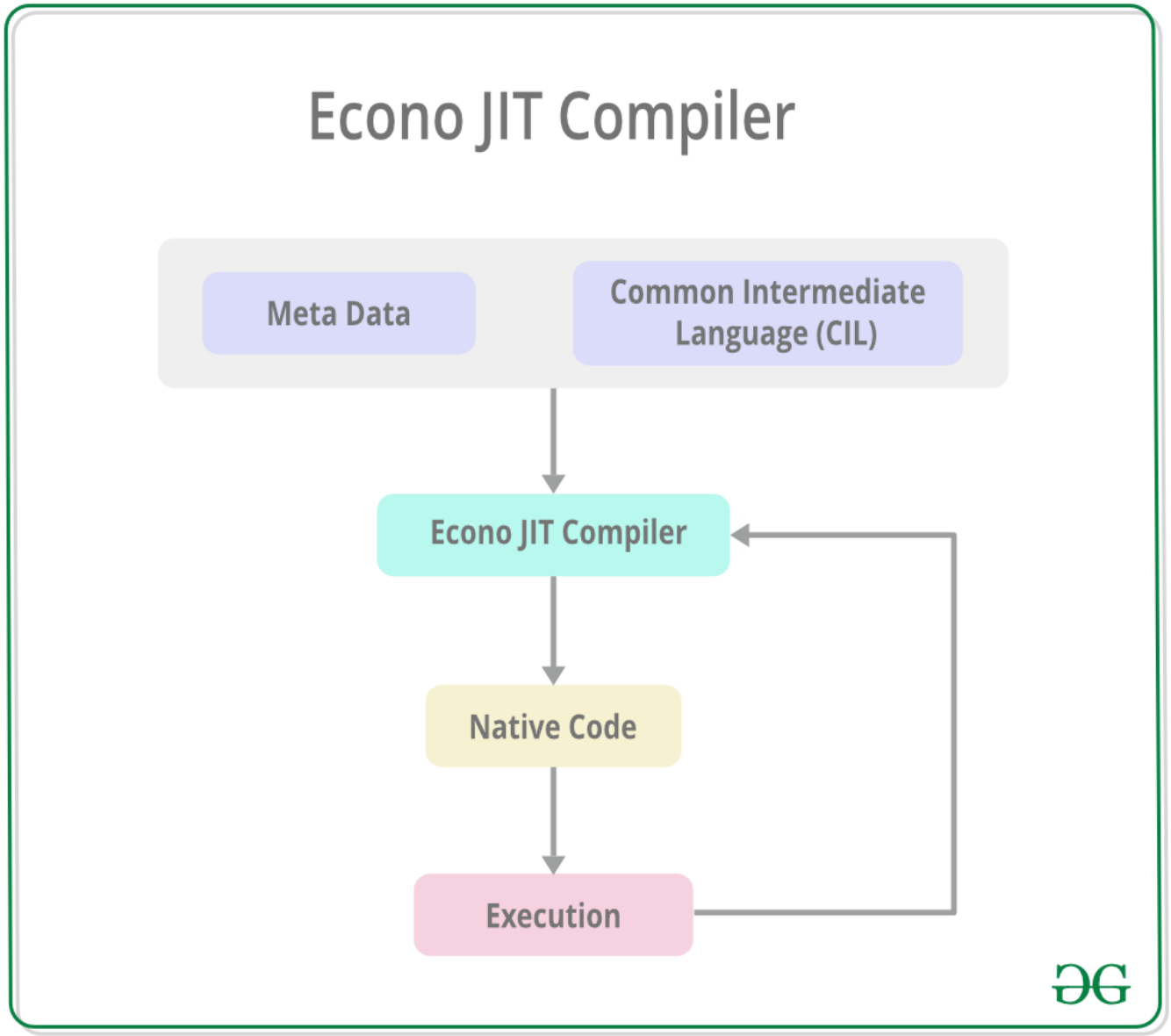
Pre-JIT Compiler: All the source code is compiled into the machine code at the same time in a single compilation cycle using the Pre-JIT Compiler. This compilation process is performed at application deployment time. And this compiler is always implemented in the ***Ngen.exe (Native Image Generator)***.



•**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.



•**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. This JIT compiler is obsolete starting from dotnet 2.0



Advantages of JIT Compiler:

•The JIT compiler requires less memory usage as only the methods that are required at run-time are compiled into machine code by the JIT Compiler.

•Page faults are reduced by using the JIT compiler as the methods required together are most probably in the same memory page.

•Code optimization based on statistical analysis can be performed by the JIT compiler while the code is running.

Disadvantages of JIT compiler:

•The JIT compiler requires more startup time while the application is executed initially.

•The cache memory is heavily used by the JIT compiler to store the source code methods that are required at run-time.

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.

.Net Framework Design Principle:

1) **Interoperability** – The .Net framework provides a lot of backward support. Suppose if you had an application built on an older version of the .Net framework, say 2.0. And if you tried to run the same application on a machine which had the higher version of the .Net framework, say 3.5. The application would still work. This is because with every release, Microsoft ensures that older framework versions gel well with the latest version.

2) Portability – Applications built on the .Net framework can be made to work on any Windows platform. And now in recent times, Microsoft is also envisioning to make Microsoft products work on other platforms, such as iOS and Linux.

3) Security – The .NET Framework has a good security mechanism. The inbuilt security mechanism helps in both validation and verification of applications. Every application can explicitly define their security mechanism. Each security mechanism is used to grant the user access to the code or to the running program.

4) Memory management – The Common Language runtime does all the work or memory management. The .Net framework has all the capability to see those resources, which are not used by a running program. It would then release those resources accordingly. This is done via a program called the “Garbage Collector” which runs as part of the .Net framework. The garbage collector runs at regular intervals and keeps on checking which system resources are not utilized, and frees them accordingly.

5) Simplified deployment – The .Net framework also have tools, which can be used to package applications built on the .Net framework. These packages can then be distributed to client machines. The packages would then automatically install the application.

VB.NET:

VB.NET is a simple, multi-paradigm object-oriented programming language designed to create a wide range of Windows, Web, and mobile applications built on the **.NET Framework**.

What is VB.NET?

- The VB.NET stands for Visual Basic. Network Enabled Technologies.

- It is a simple, high-level, object-oriented programming language developed by Microsoft in 2002.

- It is a successor of Visual Basic 6.0, that is implemented on the Microsoft .NET framework. It supports the OOPs concept, such as abstraction, encapsulation, inheritance, and polymorphism. Therefore, everything in the VB.NET language is an object, including all primitive data types (Integer, String, char, long, short, Boolean, etc.), user-defined data types, events, and all objects that inherit from its base class.

- It is not a case sensitive language, whereas, C++, Java, and C# are case sensitive language.

- Applications built using the VB.NET language are very reliable and scalable, relying on the .NET Framework to access all libraries that help to execute a VB.NET program. With this language, you can develop a fully object-oriented application that is similar to an application created through another language such as C++, Java, or C#.

- The VB.NET language is designed in such a way that any new beginner or novice and the advanced programmer can quickly develop a simple, secure, robust, high performance of web, windows, console, and

VB.NET Features

As we know, it is a high-level programming language with many features to develop a secure and robust application. These are the following features that make it the most popular programming language.

•It is an object-oriented programming language that follows various oops concepts such as abstraction, encapsulation, inheritance, and many more. It means that everything in VB.NET programming will be treated as an object.

•This language is used to design user interfaces for window, mobile, and web-based applications.

•It supports a rapid application development tool kit. In which a developer does not need to write all the codes as it can get various code automatically from its libraries. For example, when we create a form in Visual basic.net, it automatically calls events of various form in that class.

•It is not a case sensitive language like other languages such as C++, java, etc.

•It supports Boolean condition for decision making in programming.

•It also supports the multithreading concept, in which you can do multiple tasks at the same time.

•It provides simple events management in .NET application.

•A Window Form enables us to inherit all existing functionality of form that can be used to create a new form. So, in this way, it reduced the code complexity.

- It uses an external object as a **reference** that can be used in a VB.NET application.
- Automatic initialized a garbage collection.
- It follows a structured and extensible programming language for error detection and recovery.
- Conditional compilation and easy to use generic classes.
- It is useful to develop web, window, and mobile applications.

Advantages of VB.NET

•The VB.NET executes a program in such a way that runs under CLR (Common Language Runtime), creating a robust, stable, and secure application.

•It is a pure object-oriented programming language based on objects and classes. However, these features are not available in the previous version of Visual Basic 6. That's why Microsoft launched VB.NET language.

•Using the Visual Studio IDE, you can develop a small program that works faster, with a large desktop and web application.

•The .NET Framework is a software framework that has a large collection of libraries, which helps in developing more robust applications.

•It uses drop and drag elements to create web forms in .NET applications.

•However, a Visual Basic .NET allows to connect one application to another application that created in the same language to run on the .NET framework.

•A VB.NET can automatically structure your code.

•The Visual Basic .NET language is also used to transfer data between different layers of the .NET architecture such that data is passed as simple text strings.

•It uses a new concept of error handling in the Visual Basic .NET Framework. The new structure is the try, catch, and finally method used to handle exceptions as a unit. In addition, it allows appropriate action to be taken at the

Disadvantages of VB.NET

1.The VB.NET programming language is unable to handle pointers directly. Because in this language, it requires a lot of programming, and it is not easy to manage every address by a pointer. Furthermore, additional coding takes extra CPU cycles, that increases the processing time. It shows the slowness of the VB.NET application.

2.The VB.NET programming is easy to learn, that increases a large competition between the programmers to apply the same employment or project in VB.NET. Thus, it reduces a secure job in the programming field as a VB.NET developer.

3.It uses an Intermediate Language (IL) compilation that can be easily decompiled (reverse engineered), but there is nothing that can prevent an application from disintegrating

4.Just-In-Time (JIT) compiler: It is the process through which a computer can interpret IL (intermediate language) compilation and is also required to run your application. It means that the target computer needs a JIT compiler to interpret a source program in IL, and this interpretation requires an additional CPU cycle that degrades the performance of an application.

5.It contains a large collection of libraries for the JIT compiler that helps to interpret an application. These large libraries hold a vast space in our system that takes more computing time.

Prerequisite

Before learning the VB.NET, you must have the basic knowledge of Visual Basic and OOPs.

Difference Between VB.NET and Visual Basic:

VB.NET

VB.NET is also known as **Visual Basic.NET**. It stands for **Visual Basic .Network Enabled Technologies.** It is a simple, high-level, object-oriented programming language developed by Microsoft in 2002. It is a successor of Visual Basic 6.0, which is implemented on the Microsoft .NET Framework
. With this language, you can develop a fully object-oriented application that is similar to an application created through another language such as C++, Java, or C#.

Feature of VB.NET

- Inheritance (object-oriented language)
- Delegates and events
- Parameterized constructors
- Method overloading/overriding
- Type-safe
- Delegates and events

Visual Basic

Visual Basic (VB) is a programming language developed by Microsoft in 1992. The purpose of this language is to develop an application that can run on different versions of the Windows operating system.

A Visual Basic evolved from Basic Language; Basic language is easier to read than other languages.

The final version of Visual Basic was released in 1998. Microsoft then launched a Visual Basic DotNet ('VB.NET') language, which is much better than Visual Basic in all aspects such as performance, reliability, working environment, easy to build, and debugging an application.

Features of Visual Basic

•User Interface design

•Rapid Application Development

•Using this language, you can use internet or intranet services in your application.

•It has powerful database access tools, by which you can easily develop front end applications.

•It also supports ActiveX technology, in which you can access the features of other application in system application. For example: Microsoft Word, Microsoft Excel, etc.

Difference Between VB. NET and Visual Basic:

VB .NET	Visual Basic
It stands for Visual Basic. Network Enables Technology. It is also developed by Microsoft, and this language was based on the .Net Framework. Furthermore, it is specially designed for VB developers.	It is a programming language developed by Microsoft for the fastest development of a window-based operating system as well as applications.
It is a modern, fully object-oriented language that replaced VB6.	VB is the predecessor of VB.NET and was not an object-oriented language. So, it is not actively maintained.
A VB.NET uses the Common Language Runtime (CLR) component of .Net Framework at runtime. It has better features and design implementation as compared to VB-Runtime.	Visual Basic uses the VB-Runtime environment.
It is a compiled language	It is an Interpreter based language
It does not support backward compatibility.	It supports backward compatibility.
It is a type-safe language.	It is not a type-safe language.
In VB.NET, data is handled using the ADO.net protocol.	Data Connectivity and handling are done through DAO, RDO, and ADO (ActiveX Data Object) protocol,
Object does not support default property.	The Object support default property of virtual basic.
In the VB.Net parameter are passed by a default value.	In VB, most of the parameters are passed by reference.
A Multithreaded application can be developed in	It does not support the multithread concept.