Application Development Tools

Text Book

Professional C# and .NET by Wrox Publication

Chapter 1

Introduction to .NET Framework

Definitions

- **Platform**
- **Framework**

What is .NET ?

Microsoft.NET is a Framework

- .NET supports **Language interoperability** between services of various programming languages.

What is .NET ?

- **Objectives of .NET Framework**
  - Provide **object-oriented programming** environment
  - Provide environment for developing various types of applications, such as **Windows-based applications and Web-based applications**
  - Code Reusability
  - Platform Independence
  - Supports multiple languages

Languages used to develop .Net Application

- ✔ Visual C#
- ✔ VB.Net
- ✔ VC++.Net
- ✔ F#
- ✔ JavaScript
- ✔ Java
- ✔ HTML
- ✔ Python
- ✔ Ruby
- ✔ PowerShell
- ✔ SQL
- ✔ JSON
- ✔ XML
- ✔ PHP
- ✔ Go
- ✔ R
- ✔ Perl

Types of Applications

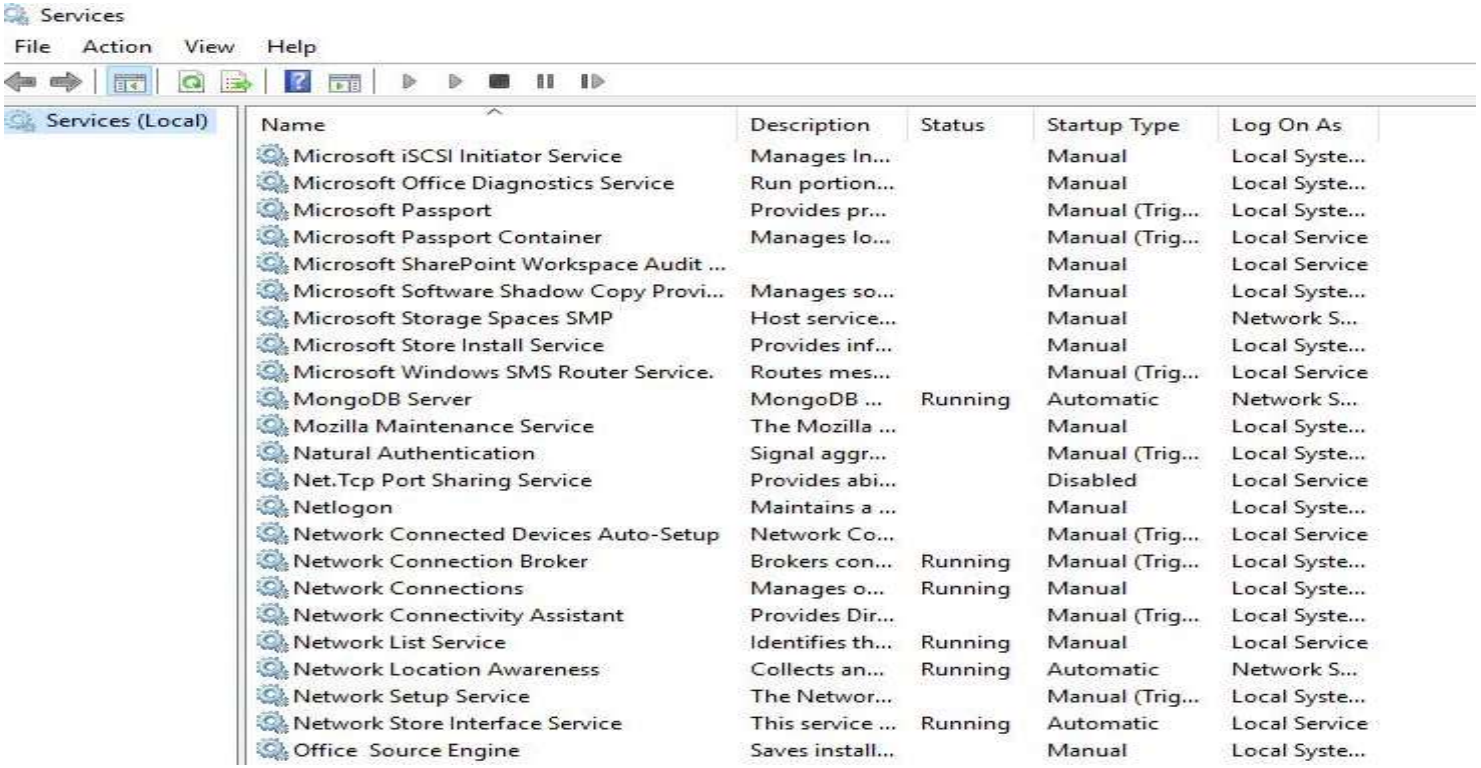
- ✔ Console Applications
- ✔ Windows Services
- ✔ Client/ Desktop Applications
  - Windows Forms
  - WPF
- ✔ Server/ Web Applications
  - ASP.NET Web Forms
  - ASP.NET MVC
  - XML Web Services
  - WCF Services
- ✔ Mobile/Smart Device Applications(Not for Android and Iphone)

1.Console application

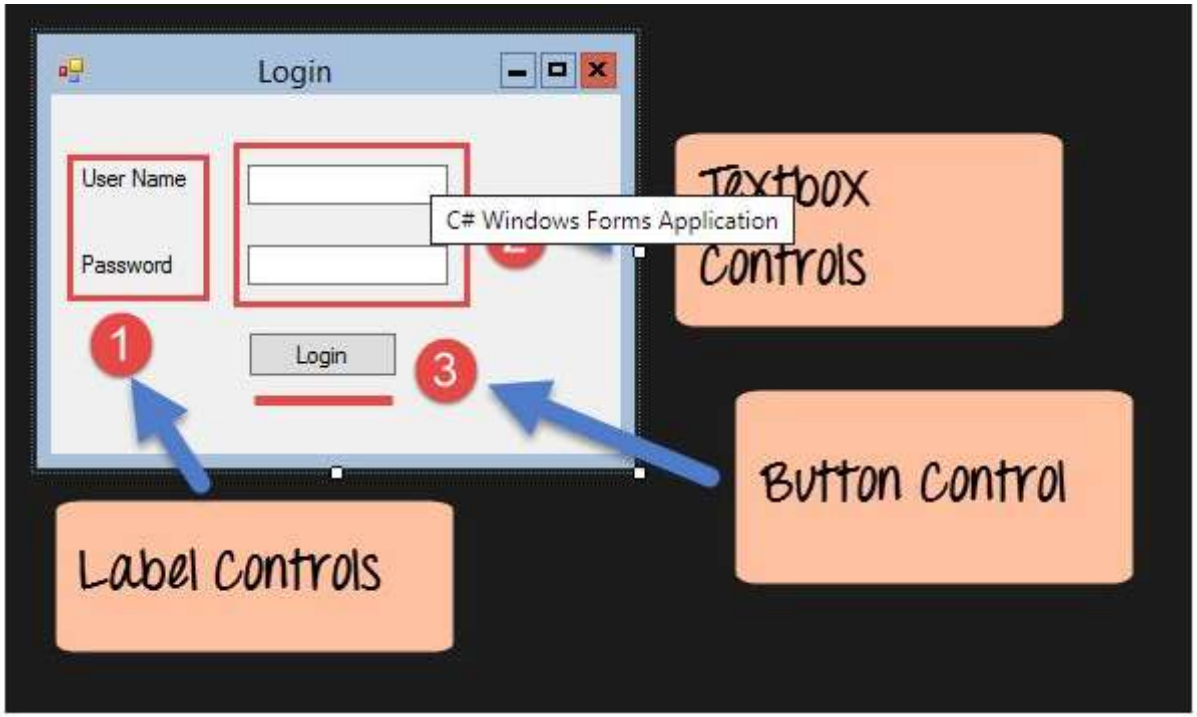
✔A console application, is an application that takes input and displays output at a command line console with access to three basic data streams: standard input, standard output and standard error.

2. Windows Services

✔WS don’t have their own user interface , they start after os started and continue run in in background. Example: SQL SERVER,IIS SERVER.(SERVER KIND OF APPLICATIONS)

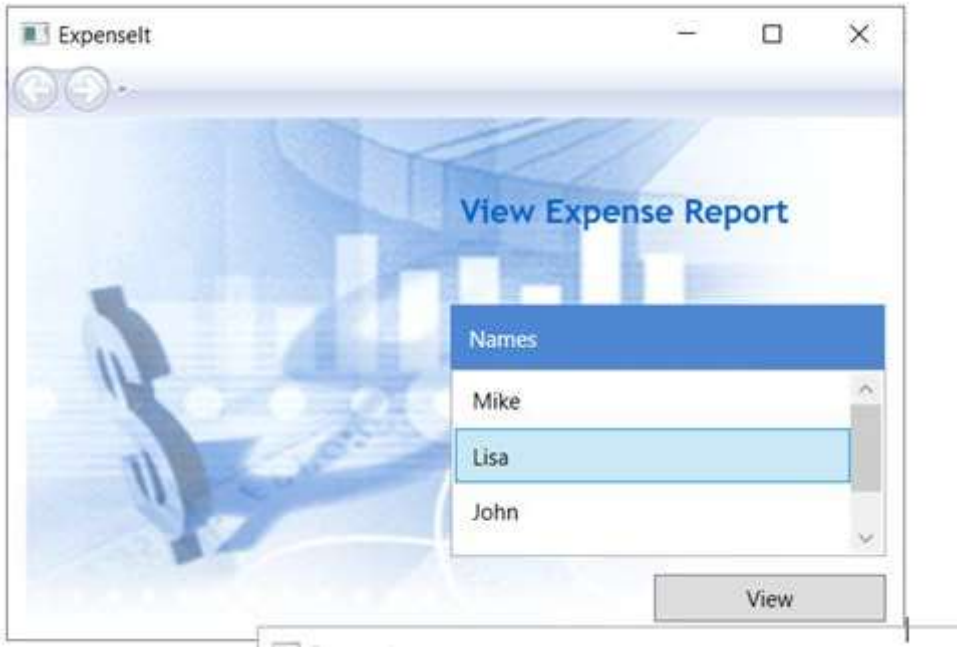


3.Windows Form Based Application



4.Windows Presentation Foundation

- supports 3D graphics, Data binding **,very interactive ,highly animated ,with lots of color compare to Windows-based applications.**



5.ASP.NET Application(Model View Controller)

- MVC patterns separate input, processing and output of an application
- **Model**: It should be responsible for the **data of the application.(Database)**
- **View**: Responsible for the **user interface which is displayed to the user.**
- **Controller**: **it takes user input, manipulates the model & causes the view to update**

6.Windows communication Foundation(WCF)

- Used for **service-oriented** applications.
- Used for are developing **components which is reusable** by different people across the network without becoming the owner of the components.

. Net Framework Architecture

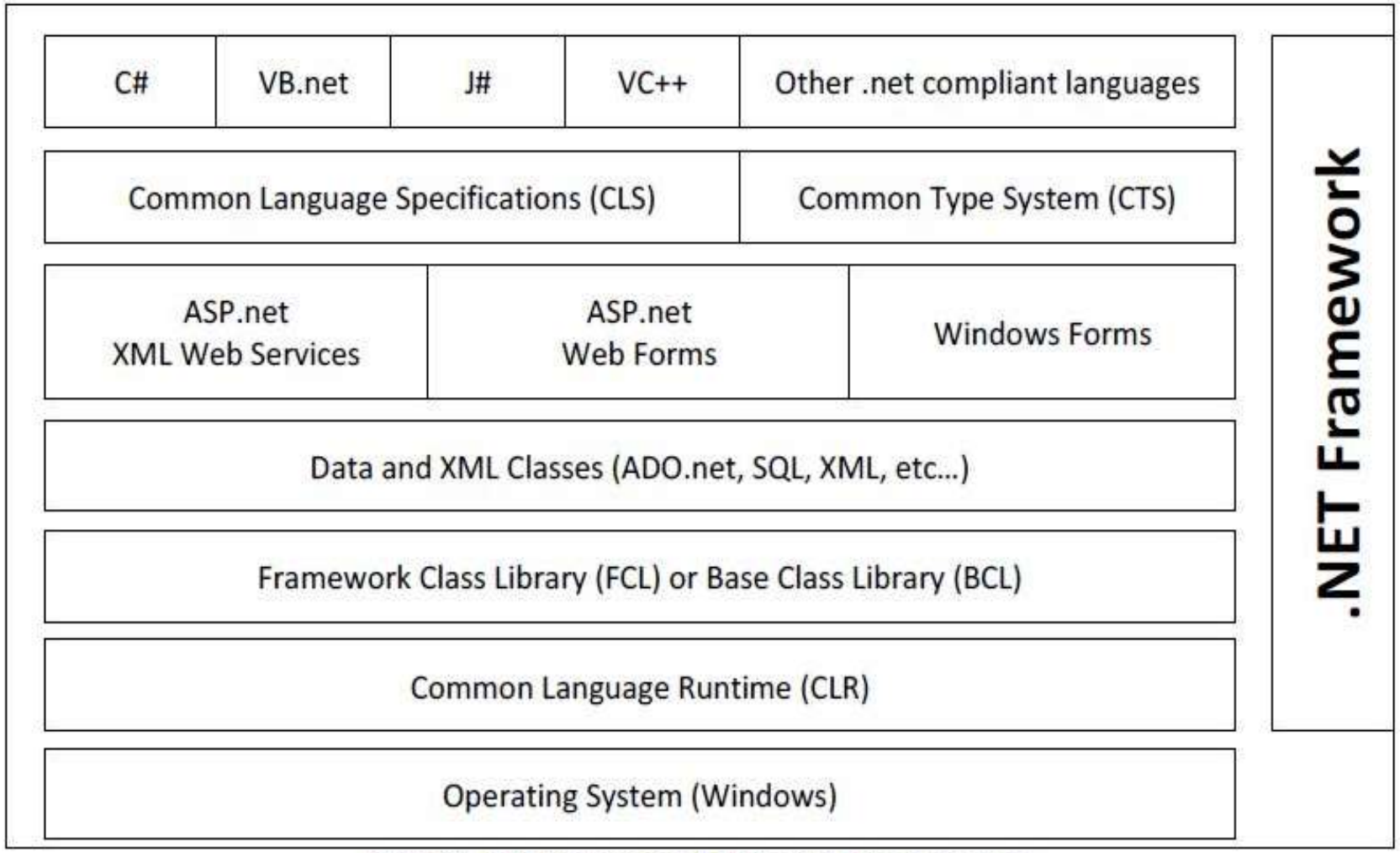


Figure1: .NET Framework Architecture Block Diagram

Components of .NET Framework

1. Common Language Runtime (CLR)
2. .Net Framework Class Library (FCL)
3. Common Type System (CTS)
4. Common Language Specification (CLS)

1. Common Language Runtime (CLR)

- It is a run-time environment which executes the code written in any .NET programming language.
- .Net framework provides the support for many languages like C#, F#, C++, Cobra, Jscript.Net, VB.Net etc.
- The code which runs under the CLR is called as **Managed Code**.
- **Functions of the CLR:**
  - It converts **program into native code**.
  - Handles Exceptions, Thread execution, Provides **type safety**
  - Automatic Memory management
  - Provides security
  - Improved performance
  - Garbage collection

Garbage collector (GC)

The garbage collector (GC) manages the **allocation and release of memory**. The garbage collector serves as an automatic memory manager.

Garbage Collection in C# has the following advantages:

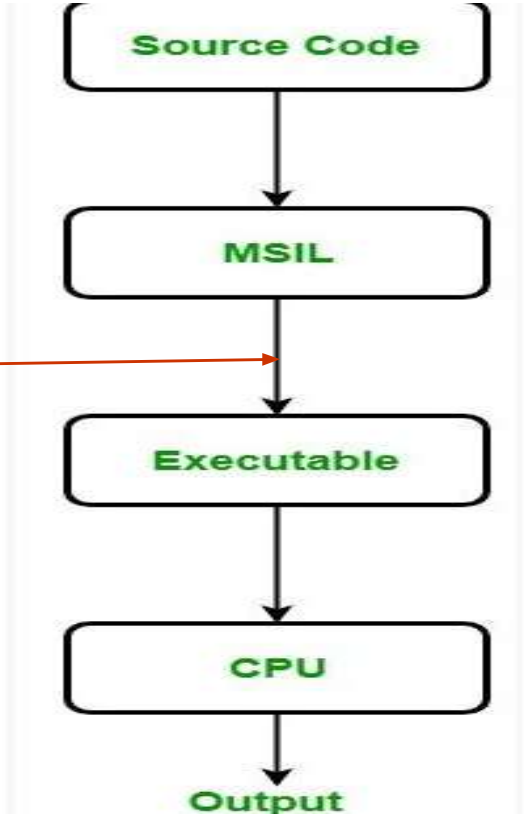
- You **don't need to free memory manually** while developing your application.
- It also **allocates objects on the heap** efficiently.
- It will **release the memory which is occupied by unused** objects.
- **Managed objects automatically get clean** by garbage collector.

Managed Code and Unmanaged Code

- **Managed code**

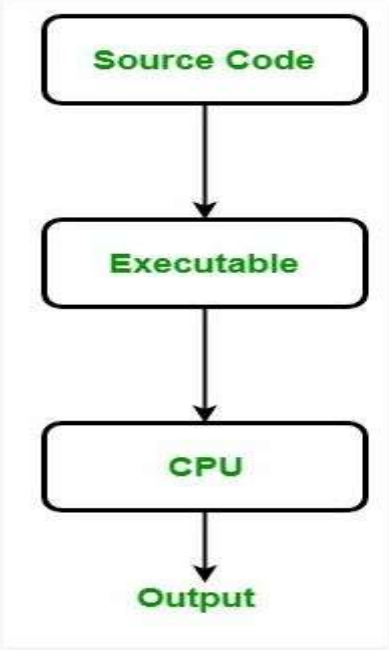
A code which is written to get the services of the runtime environment **CLR(Common Language Runtime)** in .NET Framework is known as **Managed Code**.

CLR



UnManaged code

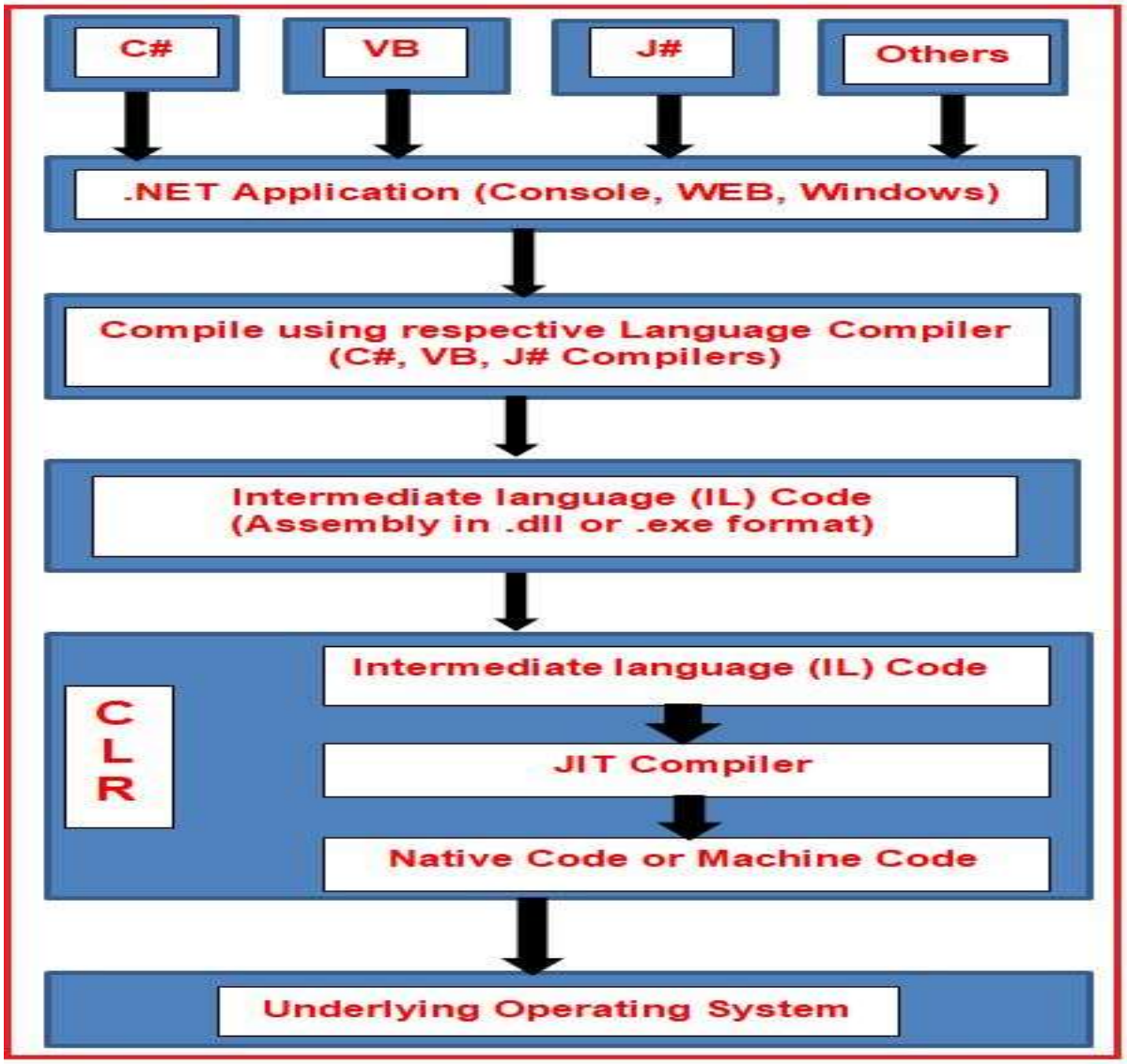
- A code which is **directly executed by the** operating system is known as **Unmanaged code**.
- After compilation it always converted into **native code that is specific to the architecture**.
- The application written in VB 6.0, C, C++, etc unmanaged code



Just In Time (JIT) Compiler:

- At execution time, a just-in-time (JIT) compiler translates the MSIL into native code.
- **Native code** refers to programming **code** that is configured to run on a specific processor.
- **JIT Types:**
  1. **Pre-JIT Compiler**: **Compiles entire code into native code** completely
  2. **Econo JIT Compiler**: complies **only those methods that are called at runtime.** However, these complied methods are removed when they are not required.
  3. **Normal JIT Compiler**: Compiles only that part of code which is called **and places in cache.When again called then use from cache.**

Execution Process



IL=MSIL = CIL

- Microsoft Intermediate Language (MSIL) is a language used as the output of a **number of compilers** (C#, VB, .NET, and so forth).
- Microsoft Intermediate Language (MSIL) is a **CPU-independent set of instructions that can be efficiently converted to the native code.**
- We can also call it as **Intermediate Language (IL)/ Common Intermediate Language (CIL)/Microsoft Intermediate language**

.

MSIL = CIL = IL(Cont..)

- The MSIL code includes **instruction to load, initialize and invoke methods on objects**.
- It also includes the instructions for **various operations on program code, such as arithmetic and logical operations, control flow, memory access, exception handling etc**.
- The program's source code is converted to MSIL code, which is equivalent to **JAVA Byte Code**.

- MSIL code is executed by the **JIT Compiler to generate native code.**
- The native code is executed by the **computer's processor**.

2. Common Type System(CTS)

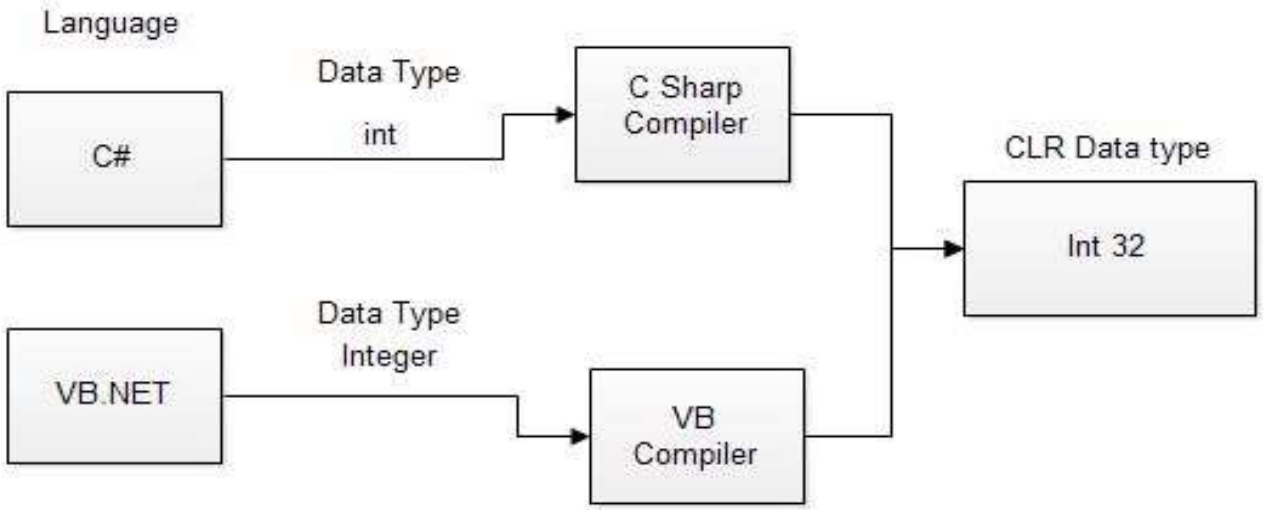
- In .NET, **every Data Type is internally represented by a class** or **structure** in CTS.

- All the **classes and structures related to Data Types are collectively** known as CTS.

- Every language provides its **own keywords for Data Types** but **internally all the languages which run under .NET framework use the classes and structures available in CTS.**

2. Common Type System(CTS) Cont..

- **For example,** when we declare an **int type data type in C#** and **VB.Net** then **after compilation** they are converted to **int32**. In other words, now both will have a common data type **Int32.**



2. Common Type System(CTS)

- The CTS performs the **following functions:**

1. It enables **cross-language integration, type safety.**

2. It **defines rules that every language must follow which runs under .NET** framework.
3. It **ensures that code written in different** .NET Languages like C#, VB.NET, F# etc. **can interact with each other**.

3.Common Language Specification

- It is a **sub set of CTS.**
- It **specifies a set of rules that needs to be satisfied by all language compilers** targeting CLR.
- It helps in **cross language inheritance and cross language debugging.**
- Provides **guidelines that language to follow so that it** can communicate with other .NET languages in a seamless manner.

3.Common Language Specification

- **Example:**one rule is that **you cannot use multiple inheritance within .NET Framework.** As you know C++ supports multiple inheritance but; when you will try to use **that C++ code within C#,** it is not possible because C# doesn't supports multiple inheritance.

- **Example :Y**ou cannot have members with same name with case difference only i.e. you cannot have **add() and Add()** methods. This easily works in **C# because** it is **case-sensitive** but when you will try to use that **C# code in VB.NET**, **it is not possible** because **VB.NET is not case-sensitive.**

4. .NET Framework Class Library

Namespace

- A namespace **is a logical collection of classes with a unique name.**
- **Example** System

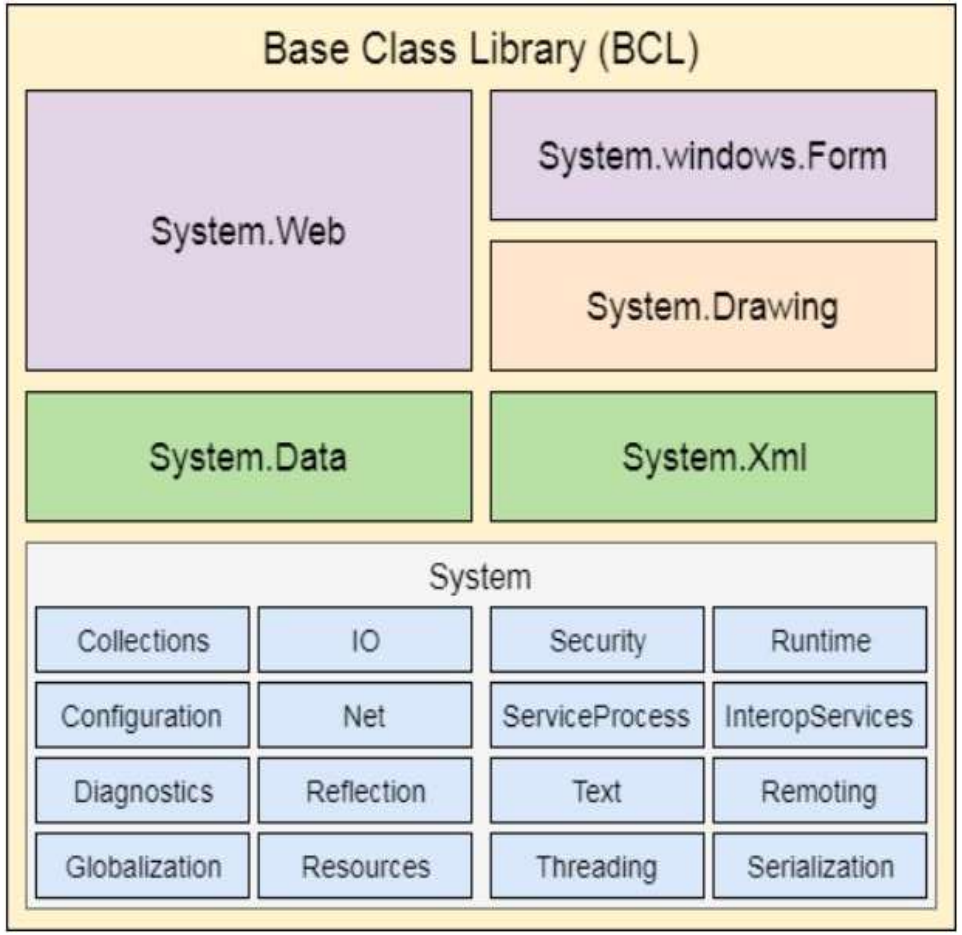
.NET Framework Class Library

- .NET Framework Class Library is the **collection of classes, namespaces, interfaces and value types** that are used for .NET applications.
- It contains **thousands of classes** that supports the following functions.
  1. Base and user-defined(class, struct etc) data types
  2. Support for **exceptions handling**
  3. **input/output** and stream operations
  4. **Communications** with **the underlying system**
  5. Access to data
  6. Ability to create **Windows-based GUI applications**
  7. Ability to create **web-client and server applications**
  8. Support for **creating web services**

.NET Framework Class Library

.NET Framework Base Class Library:

- **sub part of the Framework**
- **Provides library support** to Common Language Runtime to ***work properly***.
- Base Class library provides **classes and types that are helpful in performing day to day** operation e.g. **dealing with string and primitive types, database connection,** IO operations.



Advantages of Base Class Library

- Language Independent.
- Completely object oriented.
- BCL is included in .Net Framework
- Well optimized for performance.

Assembly

- An assembly is a file that is **automatically generated** by the compiler.
- Dynamic Link Library or an executable file. (.exe or .dll)
- **Generated only once, later assembly gets updated.**
- **Background process of your application**.
- Two types of assemblies:
  - Single file
  - Multi file
- Single file assembly: contains all the required information in a single package
  - IL
  - Metadata
  - Manifest

Assembly

- An Assembly contains three files.

Assembly

- **IL: Intermediate Language(**IL) is a Platform independent set of instructions into which all .Net languages are compiled. In order to execute application, IL code is further compiled at run time and generates machine code by JIT.
- **Metadata:**
  - Data about the data.
  - It provides
  - **complete description of all the types of information available in that assembly. Example classes, interfaces, enums, structs, namespaces, methods and their scope, and each method's parameters, type's parameters, and so on.**
- **Manifest:** Specifies assembly name,version,unique identifier,location,dependencies.

Microsoft Visual Studio

- Microsoft has introduced **Visual Studio**, which is a tool (also called Integrated Development Environment) for developing .NET applications by using programming languages such as **VB, C#, and J#.** etc.

END