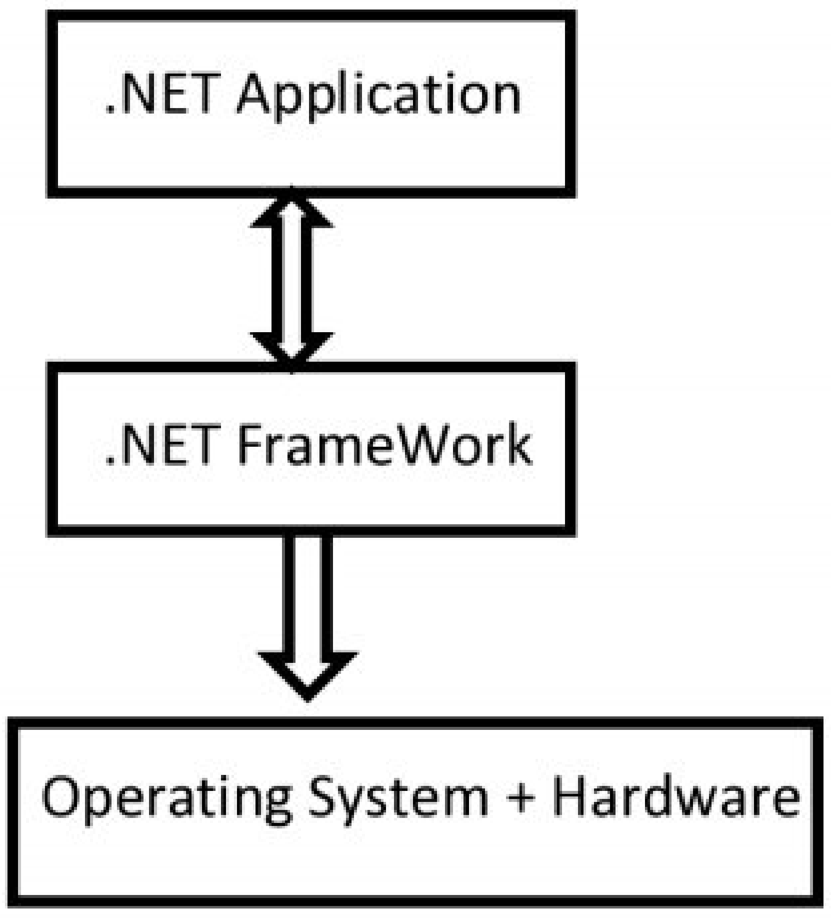
Unit 1. Overview of Microsoft .NET Framework

1.1. The .NET Framework
1.1. Managed Code MSIL, Metadata and JIT Compilation, Automatic Memory Management.
1.2. The Common Language Runtime (CLR)
1.3. The .NET Framework class Library

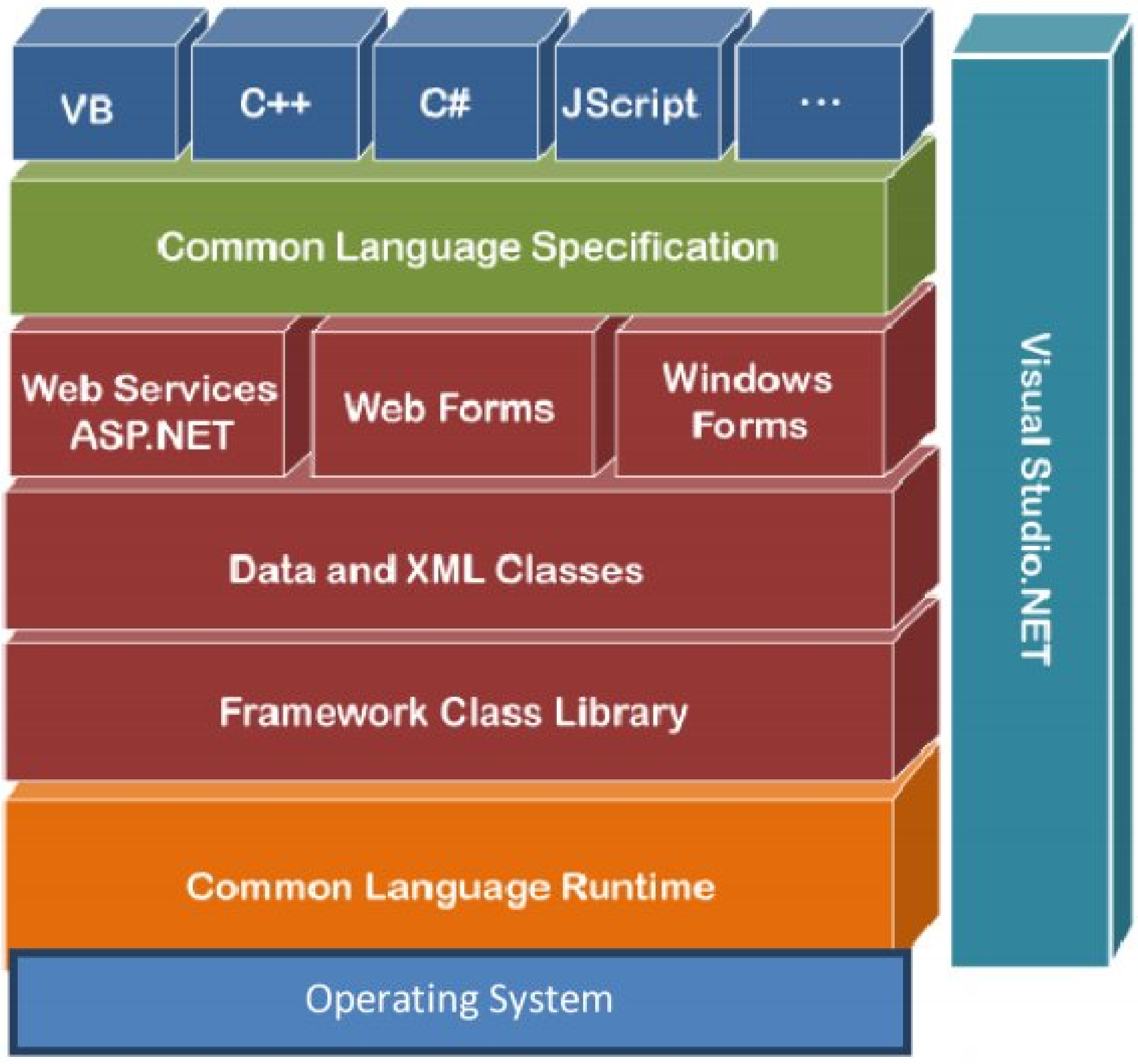
What is VB.NET?

- The VB.NET stands for Visual Basic. Network Enabled Technologies.
- .NET is a Framework or we can say “Software Platform”. It comes with many rich set of tools like Visual Studio to fully develop and build those applications.



1.1 .NET Framework?

- Framework is a set of common prefabricated software building blocks that programmer use, extend or customize for specific computing solution.
- 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.
- 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.



Components of .NET Framework

Now let's see each blocks of the Framework one bye one. In the above diagram if we see from bottom to top the first block is an Operating System. The Operating System is the system software which is installed in every computer to operate it and all other software work on it.

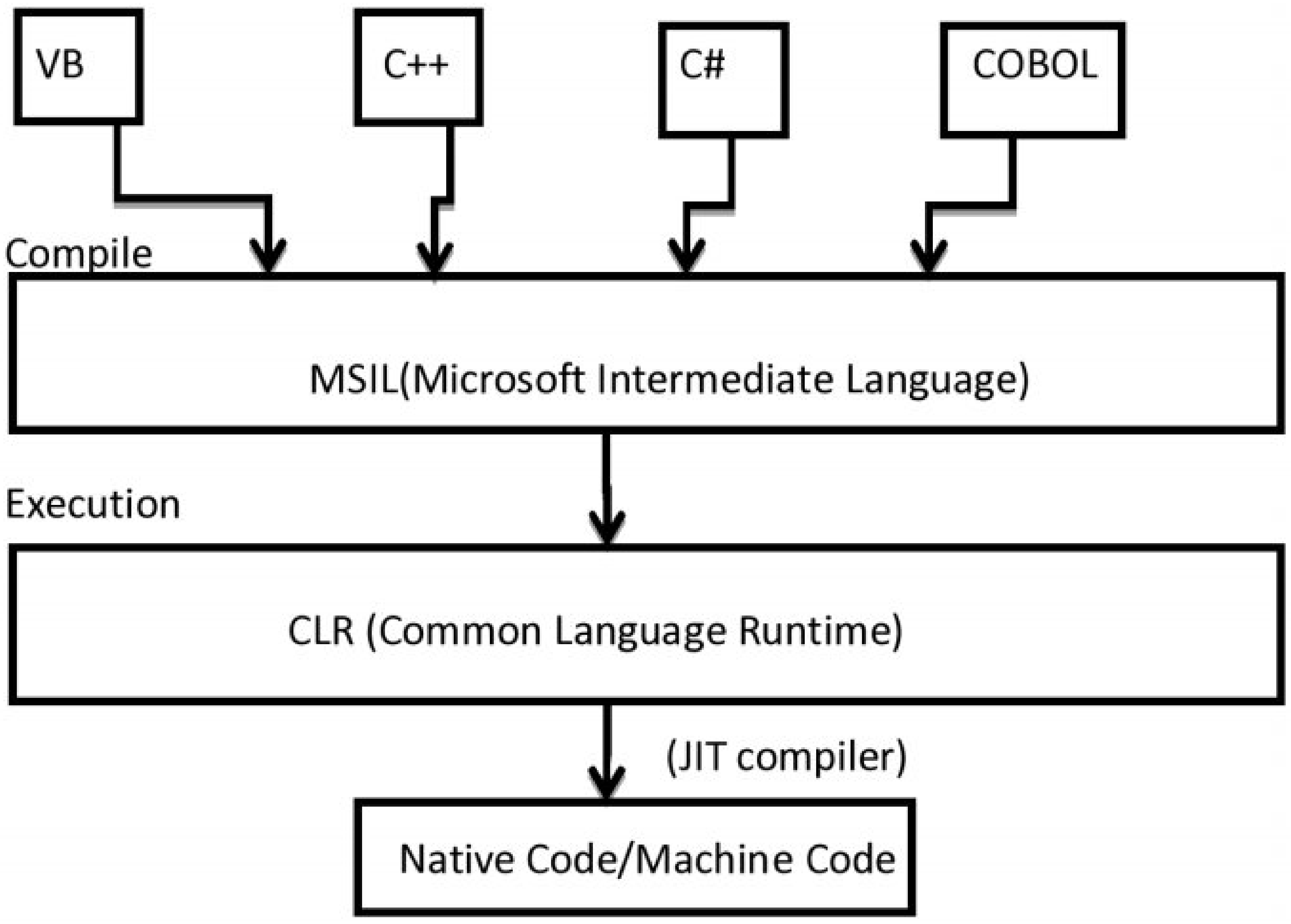
There are following components of .NET Framework:

1. CLR (Common Language Runtime)
2. BCL (Base Class Library) / FCL(Framework Class Library)
3. CLS (Common Language Specification)
4. ADO.NET and XML
5. Windows Forms
6. Web Forms & Web Services
7. CTS (Common Type System)

1. 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.

❖ **Manage Code MSIL**


VB
C++
C#
COBOL
Compile
MSIL(Microsoft Intermediate Language)
Execution
CLR (Common Language Runtime)
(JIT compiler)
Native Code/Machine Code

- Managed code is just that: code whose execution is managed by a runtime(by CLR).
- In this case, the runtime is called the **Common Language Runtime** or CLR.
- CLR is in charge of taking the managed code, compiling it into machine code and then executing it.

2. BCL (Base Class Library)

- It is also known as Framework Class Library (FCL).
- It is available to all language using .NET.
- 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.
- 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.

3. 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 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 has three levels of CLS:
  1) Complain Provider
  2) Consumer
  3) Extender

  1) Complaint Provider: Component developed by this type of language can beused by any other languages.
  2) Consumer: The language in this category can use the class developed by anyother languages.
  In simple words, the language can instantiate classesdeveloped in other languages.
  3) Extender: Languages in this category cannot just use the classes as inconsumer category; but can also extend classes using inheritance.

4. ADO.NET & XML

- It is also known as Data Access Layer.
- With the help of this layer we can access relational databases.
- It work with XML and provide the Disconnected Data Model.
- It is used to access data and data services.
- It is a part of BCL.
- It consists of two parts:
  1. Data Provider
  2. Data Set

5. Windows Forms

- It is also known as Win Forms.
- It is used to create the GUI(Graphical User Interface) for windows desktop application.
- The idea of win Form has been taken from Windows Foundation Classess(WFC) which was used for Visual J++.
- It also provides integrated and unified way of developing GUI. It has a rich variety of windows controls and user interface support. E.g. Textbox, Button, Check Box, Containers etc.
- Using Visual Studio.NET, we can simply design the GUI by dragging the controls on aform. And this is all made possible because Visual Studio.NET as it uses the"System.Winforms" namespace to draw the GUI.

6. Web Forms & Web Services

Web Forms

- It provides a tool for web application. We can say that it is a part of ASP.NET.
- It is the forms engine that provides browser-based user interface.
- It consists of two parts:
  1)Template-It contains HTML based layout information for all GUI elements.
  2)Component-It contains the logic behind the controls.
- With ASP.NET, Microsoft has provided presentation-business layer separation – by introducing the concept of Web Forms: ASP.NET Web Forms provide an easy and powerful way to build dynamic WUI (Web User Interface).

Web Services

- Web Services are the applications that run on a Web Server and communicate with other applications.
- It uses a series of XMLbased communicating protocols that respond to different requests.
  XML-(Extensible MarkupLanguage It is a standard which is used for storing, carrying and exchanging data over the internet)
- They are the small unit of code which is design to handle limited set of task.
- One example could be flight schedules and ticket reservation systems.
- The protocols on which Web Services are built summarized below:
  UDDI: Stands for Universal Discovery and Description Integration. It allowsBusinesses to search for other Businesses, allowing them to search for theservices it needs, know about the services and contact them.
  WSDL: Stands for Web Services Description Language, often called as whiz-dull. WSDL is an XML document that describes a set of SOAP messages andhow those messages are exchanged.

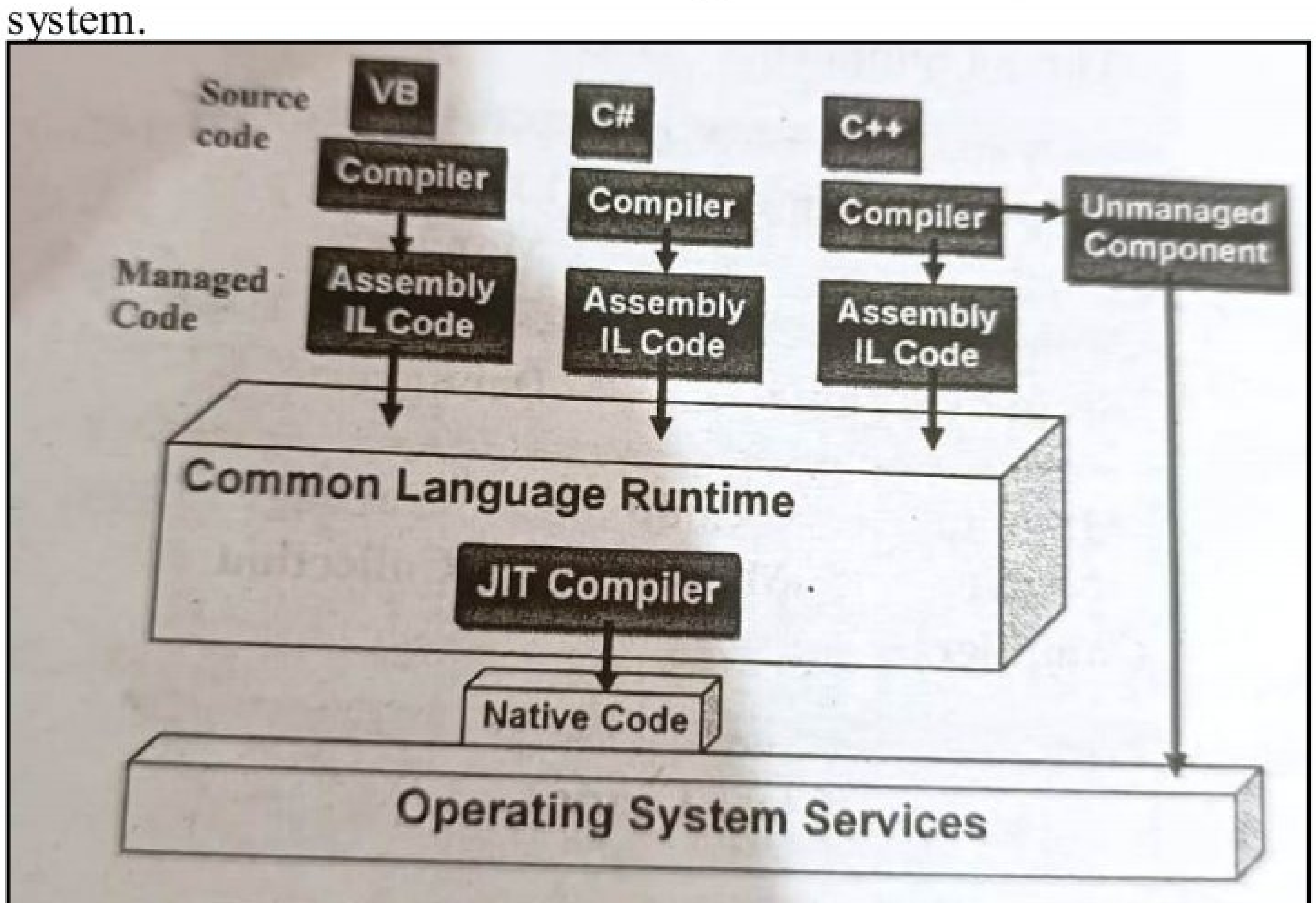
SOAP: Stands for Simple Object Access Protocol. It's the communicationprotocol for Web Services.
XML, HTTP and SMTP: Stands for Extensible Markup Language, HyperText Transfer Protocol and Simple Message Transfer Protocol respectively.UDDI, WSDL and SOAP rely on these protocols for communication.

7. **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.

1.2 The Common Language Runtime (CLR)

- It is aheart of the .NET framework.
- It is used to compile the MSIL code to the native code. Below Figure shows the execution model ofCLR.
- The common language runtime is one of the most essential components of the .NET framework.
- The CLR provides functionality such as exception handling, security,debugging and versioning support to any languages that targets it.
- The code that isdeveloped with a language compiler that targets the CLR is called **managed code**.
- Onthe other hand, the code developed outside the .NET framework is known as **unmanaged code**. Applications that do not run under the control of the CLR are said to be unmanaged. Certain languages such as C++ can be used to write such applications, such as low-level access functions of the operating system.



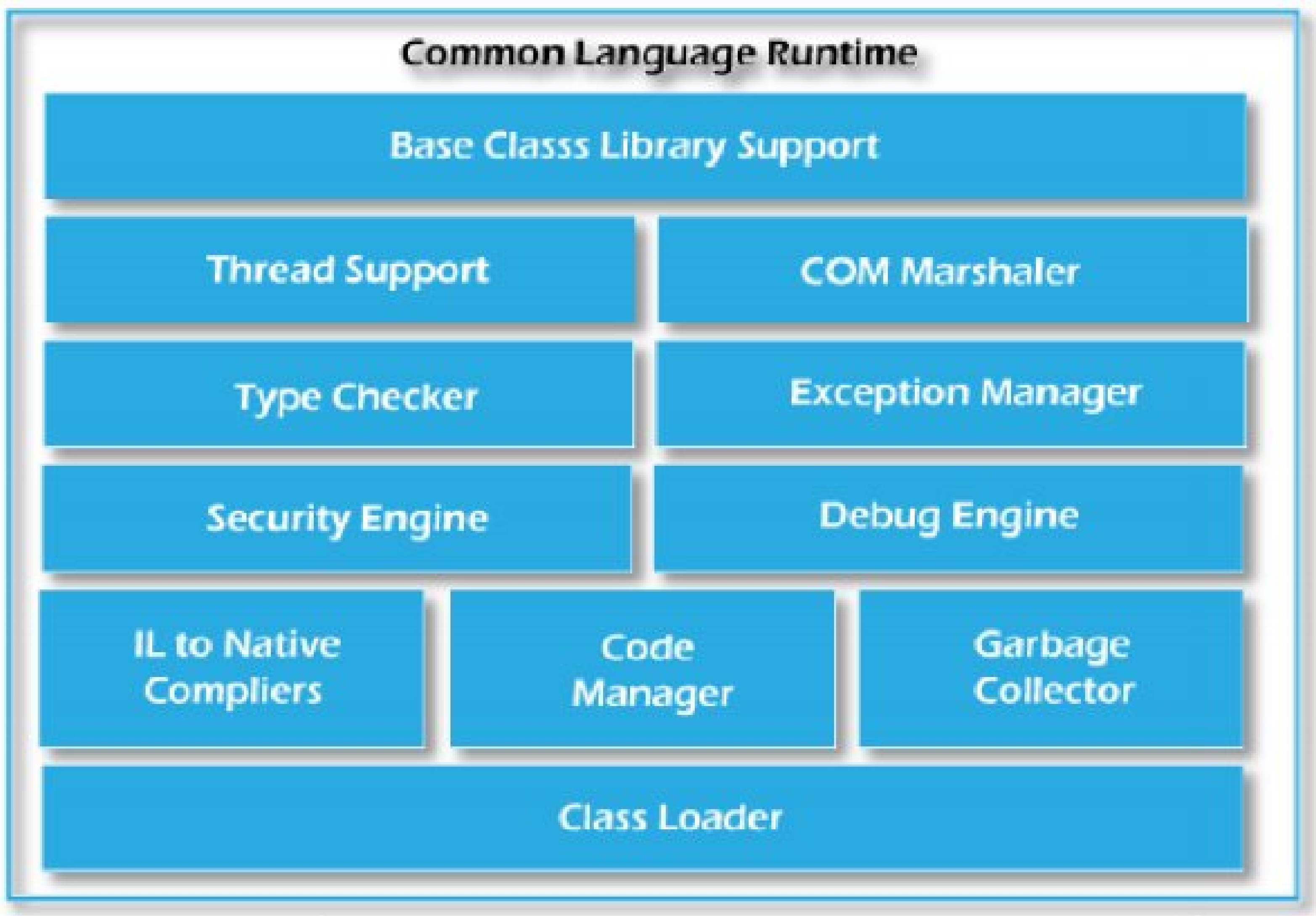
[Fig (1.e): CLR – Execution Model]

- When the .NET program is compiled, the output of the compiler is not an executable file but a file that contains a special type of code called the Microsoft Intermediate Language (MSIL), which is a low-level set of instructions understood by the common language run time.
- This MSIL is turned into executable code using a JIT(Just In Time) complier. The process goes like this, when .NET programs are executed,the CLR activates the JIT complier.
- The JIT complier converts MSIL into native codeon a demand basis (i.e. each part of the program is needed).
- Thus the program executesas a native code even though it is compiled into MSIL making the program to runfast. So, it achieves the portability benefits of MSIL.

What is JIT?

- It stands for “Just-In-Time”.
- It’s a smart compiler.
- JIT does not compile whole program each and every time. It compiles only that portion of the program in which functions are called during that time if native code is already present then that data will not again compiled. If changes are made then it will generates MSIL to native.
- Firstly, any program compile by its own compiler then it will convert into MSIL then with the help of JIT; MSIL compile into Native code but CLR does not convert whole MSIL code to Native code on load time of that application; instead it compiles the MSIL instructions as they are called.

Purpose/Role of CLR



Components of the Common Language runtime/Architecture of CLR

✓ **Class Loader**:

As and when needed, it loads the classes into the system memory.

✓ **Code Manager**:

CLR takes care of code management upon programexecution and provides various services such as memory management, threadmanagement, security management and other system services.

✓ **Garbage Collection**:

Garbage Collection is a mechanism that allows the computer to detect when an object can no longer be accessed.

It automatically free up the memory used by that object.

One of the advantages of CLR is automatic memory management that uses the Garbage Collection mechanism.

The CLR’s Garbage Collector(GC) manages the allocation and release of memory for an application.

✓ **MSIL to Native Compilers**:

When you compile a program developed in alanguage that targetstheCLR, the compiler translates the code into anintermediate language.

This language is CPU-independent. This means thatthe code can be executed on any platform that supportsthe .NET CLR as itconverts the MSIL code to Native code.

✓ **Debugger**:

It performs the debugging at runtime.

✓ **Security Engine**:

In .NET platform, security is achieved through the Code Access Security(CAS) model.

In this model, the CLR enforces restrictions onmanaged code through the use of objects called 'permissions'.

In otherwords, the CAS model specifies what the code can access instead of specifyingwho can access resources.

✓ **Exception Manager:**

It handles all the runtime exceptions occurring duringthe execution of an application.

✓ **Type Checker**:

This feature ensures that objects are always accessed incompatible ways.
Therefore, the CLR will prohibit a code from assigning a 10-bytes value to an object that occupies 8 bytes.

✓ **Thread Support**:

Thread is a light weight process. Process is nothing but theprogram which is in execution. Thread provides the multi-threading support to an application.

✓ **CMO Marshaller:**

It allows the communication between the application and COM(Component Object Model)

✓ **Class Library Support:**

It provides BCL classes when application needs at execution time

1.3 The.NET Framework Class Library

- .NET Framework Class Library is the collection of classes, namespaces, interfaces and value types that are used for .NET applications.
- Class library is the second major entity of the .NET Framework which is designed tointegrate with the common language runtime.
- It gives the program access to runtimeenvironment.
- It consists of lots of prewritten code that all the applications created inVB.NET and Visual Studio .NET will use it.
- The code for all the elements like forms,controls and the rest in VB.NET applications actually comes from the class library.
- It is built on the object-oriented nature of the runtime.
- It provides classes that can beused in the code to accomplish a range of common programming tasks, such as stringmanagement, data collection, database connectivity, and file access.
- One of the mostimportant features of the .NET framework class library is that it can be used inconsistent manner across multiple languages.
- This means that you can use the same setof classes for performing a specific task in VB as well as in VC++. This not only makesthe .NET framework types easy to use but also simplifies the learning curve associatedwith using a new piece of code.
- The .NET framework class library comprises of namespaces, which are contained within assemblies.

1.3.1 Namespaces:

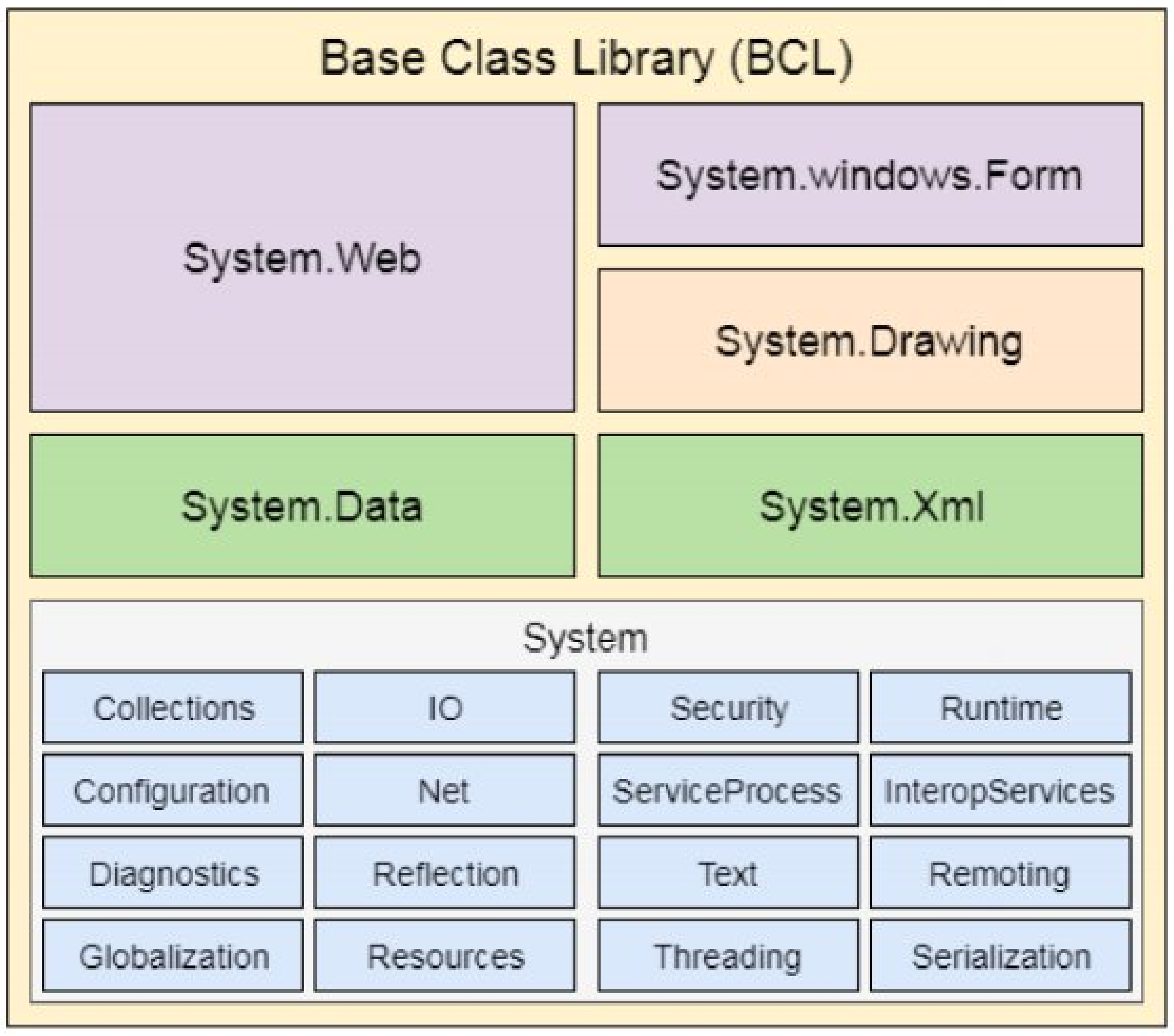
- Namespaces help you to create logical groups of related classes and interfaces thatcan be used by any language targeting the .NET framework.
- Namespaces allowyou to organize your classes so that they can be easily accessed in otherapplications. Namespace can also be used to avoid any naming conflicts betweenclasses that have the same names.
- For example, you can use two classes with thesame name in an application provided they belong to different namespaces.
- You can access the classes belonging to a namespace by simply importing thenamespace into an application.
- The .NET framework uses a dot (.) as a delimiterbetween classes and namespaces.
- E.g. System.Console represents the Consoleclass of the System namespace.
- The following are the main areas covered by Class library.

1) Data Structures
2) IO management
3) Windows and Web Controls
4) Database access
5) Multithreading
6) Remoting
7) Reflections

Common Namespaces:

Namespace	Description
System	Contains fundamental classes and base classes.
System.IO	Contains classes for reading and writing data in file.
System.XML	Contains classes work with XML
System.Windows.Forms	Contains classes for Windows based application
System.Data	Contains classes for the database connection.

It includes the System namespace and core types of the .NET framework.



1.3.2 Common Type System (CTS)

- It describes how types are declared, used and managed.
- It facilitates cross-language integration, type safety, and high performance code execution.
- It is aspecification that defines the rules to support language integration.
- This is donein such a way, that programs written in any language (.NET compliant) caninteroperate with one another.
- This also can take full advantage of inheritance,polymorphism, exceptions, and other features.
- It makes sure that all .NETapplications use the same data types.

❖ **MetaData:**

- Metadata means data about data.
- Metadata stored within the Assembly.
- .Net records information about compiled classes as Metadata.
- A .Net language compile will generate the metadata and store this in the assembly.

❖ **Assembly:**

- An assembly is the primary building block of a .Net Framework application.
- An Assembly is a logical DLL.
- It consists of DLLs or execute tables.
- It is a collection of functionality that is built, versioned, and deployed as a single implementation unit(as one or more files).
- A private assembly is used only by a single application, and is stored in that application's install directory.
- A shared assembly is one that can be referenced by more than one application.