**CSharp Notes:**

**Difference between C++ and C#**

C++ is a low level and platform neutral programming language. C# is a high-level language

C++ is a language that runs on all sorts of platforms. It is also equally popular on Unix and Linux systems. C#, while standardized, is rarely seen outside windows.

C++ is not a complete object orient language. C# is a pure object-oriented language.

C++ programmers generally focus on applications that work directly with hardware or that need better performance than other languages can offer. C# is used for modern app development.

In C++, you need to manage memory manually. C# runs memory management automatically

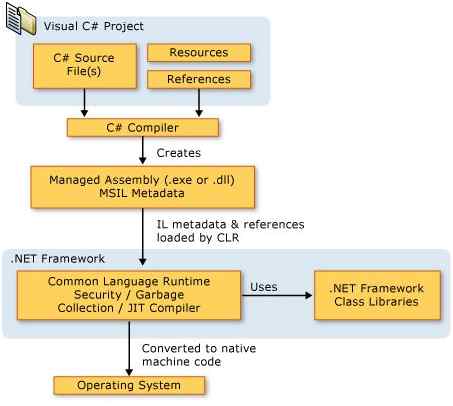
C++ does not support garbage collection. C# supports garbage collection.

What is the .NET Framework?

**Answer:**The .NET is a Framework, which is a collection of classes of reusable libraries given by Microsoft to be used in other .NET applications and to develop, build and deploy many types of applications on the Windows platform including the following:

* Console Applications
* Windows Forms Applications
* Windows Presentation Foundation (WPF) Applications
* Web Applications
* Web Services
* Windows Services
* Services-oriented applications using Windows Communications Foundation (WCF)
* Workflow-enabled applications using Windows Workflow Foundation(WF)

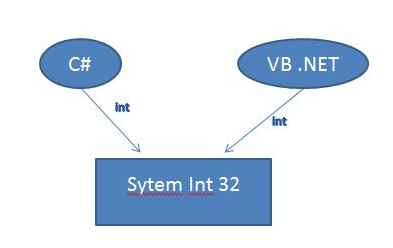
## Question 2. What is CLR?

<https://www.c-sharpcorner.com/UploadFile/puranindia/interview-question-on-net-framework-or-clr/>  
**Answer:**The CLR stands for Common Language Runtime and it is an Execution Environment. It works as a layer between Operating Systems and the applications written in .NET languages that conforms to the Common Language Specification (CLS). The main function of Common Language Runtime (CLR) is to convert the Managed Code into native code and then execute the program. The Managed Code compiled only when it is needed, that is it converts the appropriate instructions when each function is called. The Common Language Runtime (CLR)’s just in time (JIT) compilation converts Intermediate Language (MSIL) to native code on demand at application run time.  
  
When a .NET application is executed at that time the control will go to Operating System, then Operating System create a process to load **CLR.**  
The program used by the operating system for loading CLR is called runtime host, which are different depending upon the type of application that is desktop or web based application i.e.  
  
The runtime host for **desktop applications** is API function called **CorbinToRuntime**.  
  
The runtime host for **web based**applications is ASP.NET worker process **(aspnet-wp.exe)**.  
  
  
  
CLR runtime engine comes with set of services, which are classified as follows  
  
**CLR services**

* Assembly Resolver
* Assembly Loader
* Type Checker
* COM marshalled
* Debug Manager
* Thread Support
* IL to Native compiler
* Exception Manager
* Garbage Collector

To know more about them follow the link:

Question 3. What is CTS?

**Answer:**The Common Type System (CTS) standardizes the data types of all programming languages using .NET under the umbrella of .NET to a common data type for easy and smooth communication among these .NET languages.  
  
  
  
To implement or see how CTS is converting the data type to a common data type, for example, when we declare an int type data type in C# and VB.NET, then they are converted to int32. In other words, now both will have a common data type that provides flexible communication between these two languages.

Question 4. What is CLS?

**Answer**: One of the important goals of .NET Framework is to support Multiple Languages i.e VB, C#, Phyton etc. This is achieved by CLS. For multiple languages to interoperate, it is necessary that they should go on in common in certain features such as Types that are used. For example, every language has its own size and range for different data types. Thus CLS is the agreement among language designers and class library designers concerning these usage conventions.

What is managed code?

**Answer:** The resource, which is within your application domain is, managed code. The resources that are within domain are faster.  
The code, which is developed in .NET framework, is known as managed code. This code is directly executed by CLR with help of managed code execution. Any language that is written in .NET Framework is managed code.

Question 7. What is JIT?

**Answer**: A Web Service or Web Forms file must be compiled to run within the CLR. Compilation can be implicit or explicit. Although you could explicitly call the appropriate compiler to compile your Web Service or Web Forms files, it is easier to allow the file to be complied implicitly. Implicit compilation occurs when you request the .asmx via HTTP-SOAP, HTTP-GET, or HTTP-POST. The parser (xsp.exe) determines whether a current version of the assembly resides in memory or in the disk. If it cannot use an existing version, the parser makes the appropriate call to the respective compiler (as you designated in the **Class** property of the .asmx page).  
  
When the Web Service (or Web Forms page) is implicitly compiled, it is actually compiled twice. On the first pass, it is compiled into IL. On the second pass, the Web Service (now an assembly in IL) is compiled into machine language. This process is called Just-In-Time JIT compilation because it does not occur until the assembly is on the target machine.