QUHASA

. NET Interview QUESTIONS

---

1)What is a IL or What is MSIL or CIL , What is JIT?

Answer:
>(IL)Intermediate Language is also known as MSIL (Microsoft Intermediate Language) or CIL (Common Intermediate Language). All .NET source code is compiled to IL. This IL is then converted to machine code at the point where the software is installed, or at run-time by a Just-In- Time (JIT) compiler.

---

2)What is a CLR?

Answer:
>Full form of CLR is Common Language Runtime and it forms the heart of the .NET framework. All Languages have runtime and its the responsibility of the runtime to take care of the code execution of the program. For example VC++ has MSCRT40.DLL,VB6 has MSVBVM60.DLL, Java has Java Virtual Machine etc. Similarly .NET has CLR. Following are the responsibilities of CLR:-
1. Garbage Collection - CLR automatically manages memory thus eliminating memory leaks. When objects are not referred GC automatically releases those memories thus providing efficient memory management.
2. Code Access Security - CAS grants rights to program depending on the security configuration of the machine. Example the program has rights to edit or create a new file but the security configuration of machine does not allow the program to delete a file. CAS will take care that the code runs under the environment of machines security configuration.
3. Code Verification - This ensures proper code execution and type safety while the code runs. It prevents the source code to perform illegal operation such as accessing invalid memory locations etc.
IL(Intermediate language)-to-native translators and optimizerâ€™s - CLR uses JIT and compiles the IL code to machine code and then executes. CLR also determines depending on platform what is optimized way of running the IL code.

---

3)What is a CTS?

Answer:
>n order that two language communicate smoothly CLR has CTS (Common Type System). Example in VB you have "Integer" and in C++ you have "long" these datatypes are not compatible so the interfacing between them is very

complicated. In order to able that two different languages can communicate Microsoft introduced Common Type System. So "Integer" datatype in VB6 and "int" datatype in C++ will convert it to System.int32 which is datatype of CTS. CLS which is covered in the coming question is subset of CTS. Note: If you have undergone COM programming period interfacing VB6 application with VC++ application was a real pain as the datatype of both languages did not have a common ground where they can come and interface, by having CTS interfacing is smooth.

---

4)What is a CLS(Common Language Specification)?

Answer:
>This is a subset of the CTS which all .NET languages are expected to support. It was always a dream of Microsoft to unite all different languages in to one umbrella and CLS is one step towards that. Microsoft has defined CLS which are nothing but guidelines that language to follow so that it can communicate with other .NET languages in a seamless manner.

---

5)What is a Managed Code?

Answer:
>Managed code runs inside the environment of CLR i.e. .NET runtime. In short all IL are managed code. But if you are using some third party software example VB6 or VC++ component they are unmanaged code as .NET runtime (CLR) does not have control over the source code execution of the language.

---

6)What is a Assembly?

Answer:
>1. Assembly is unit of deployment like EXE or a DLL.
2. An assembly consists of one or more files (dlls, exeâ€™s, html files etc.), and represents a group of resources, type definitions, and implementations of those types. An assembly may also contain references to other assemblies. These resources, types and references are described in a block of data called a manifest. The manifest is part of the assembly, thus making the assembly self-describing.
3. An assembly is completely self-describing.An assembly contains metadata information, which is used by the CLR for everything from type checking and security to actually invoking the components methods. As all information is in the assembly itself, it is independent of registry. This is the basic advantage as compared to COM where the version was stored in registry.
4. Multiple versions can be deployed side by side in different folders. These different versions can execute at the same time without interfering with each other. Assemblies can be private or shared. For private assembly deployment, the assembly is copied to the same directory as the client program that

references it. No registration is needed, and no fancy installation program is required. When the component is removed, no registry cleanup is needed, and no uninstall program is required. Just delete it from the hard drive.
5. In shared assembly deployment, an assembly is installed in the Global Assembly Cache (or GAC). The GAC contains shared assemblies that are globally accessible to all .NET applications on the machine.

---

7)What are the different types of Assembly?

Answer:
>There are two types of assembly Private and Public assembly. A private assembly is normally used by a single application, and is stored in the application's directory, or a sub-directory beneath. A shared assembly is normally stored in the global assembly cache, which is a repository of assemblies maintained by the .NET runtime. Shared assemblies are usually libraries of code which many applications will find useful, e.g. Crystal report classes which will be used by all application for Reports.

---

8)What is NameSpace?

Answer:
>Namespace has two basic functionality :-
1. NameSpace Logically group types, example System.Web.UI logically groups our UI related features.
2. In Object Oriented world many times its possible that programmers will use the same class name.By qualifying NameSpace with classname this collision is able to be removed.

---

9)What is Difference between NameSpace and Assembly?

Answer:
>Following are the differences between namespace and assembly :-
1. Assembly is physical grouping of logical units. Namespace logically groups classes.
2. Namespace can span multiple assembly

---

10)What is Manifest?

Answer:
>Assembly metadata is stored in Manifest. Manifest contains all the metadata needed to do the following things:-
1. Version of assembly.

2. Security identity.
3. Scope of the assembly.
4. Resolve references to resources and classes.
5. The assembly manifest can be stored in either a PE file (an .exe or .dll) with Microsoft intermediate language (MSIL) code or in a stand-alone PE file that contains only assembly manifest information.

---

11) Where is version information stored of an assembly?

Answer:
>Version information is stored in assembly in manifest.

---

12)Is versioning applicable to private assemblies?

Answer:
>Versioning concept is only applicable to global assembly cache (GAC) as private assembly lie in their individual folders.

---

13)What is GAC?

Answer:
>GAC (Global Assembly Cache) is used where shared .NET assembly reside. GAC is used in the following situations :-
1. If the application has to be shared among several application.
2. If the assembly has some special security requirements like only administrators can remove the assembly. If the assembly is private then a simple delete of assembly the assembly file will remove the assembly.

---

14)What is the concept of strong names?

Answer:
>Strong name is similar to GUID(It is supposed to be unique in space and time) in COM components.Strong Name is only needed when we need to deploy assembly in GAC. Strong Names helps GAC to differentiate between two versions. Strong names use public key cryptography (PKC) to ensure that no one can spoof it.PKC use public key and private key concept.

---

15)How to add and remove an assembly from GAC?

Answer:

>There are two ways to install .NET assembly in GAC:-
1. Using Microsoft Installer Package. You can get download of installer from http://www.microsoft.com.
2. Using Gacutil. Goto "Visual Studio Command Prompt" and type "gacutil â€“i (assembly_name)", where (assembly_name) is the DLL name of the project.

---

16) What is garbage collection?

Answer:
>Garbage collection is a CLR feature which automatically manages memory. Programmers forget to release the objects while coding.(Remember in VB6 where one of the good practices is to set object to nothing). CLR automatically releases objects when they are no longer in use and refernced. CLR runs on non-deterministic to see the unused objects and cleans them. One side effect of this non-deterministic feature is that we cannot assume an object is destroyed when it goes out of the scope of a function. Therefore, we should not put code into a class destructor to release resources.

---

17)Can we force garbage collector to run?

Answer:
>System.GC.Collect() forces garbage collector to run. This is not recommended but can be used if situations arises.

---

18)What is reflection?

Answer:
>All .NET assemblies have metadata information stored about the types defined in modules. This metadata information can be accessed by mechanism called as "Reflection".System. Reflection can be used to browse through the metadata information.

---

19)What are Value types and Reference types?

Answer:
>Value types directly contain their data which are either allocated on the stack or allocated in-line in a structure. Reference types store a reference to the value's memory address, and are allocated on the heap. Reference types can be self-describing types, pointer types, or interface types. Variables that are value types each have their own copy of the data, and therefore operations on one variable do not affect other variables. Variables that are reference types can

refer to the same object; therefore, operations on one variable can affect the same object referred to by another variable. All types derive from the System.Object base type.

---

20) What is concept of Boxing and Unboxing?

Answer:
>Boxing permits any value type to be implicitly converted to type object or to any interface type implemented by value type. Boxing is a process in which object instances are created and copy values in to that instance. Unboxing is vice versa of boxing operation where the value is copied from the instance in to appropriate storage location. Below is sample code of boxing and unboxing where integer data type is converted in to object and then vice versa.

```
Dim x As Integer
Dim y As Object
x =10
y = x â€˜ boxing process
x = y â€˜ unboxing process
```
