**Intereview Q/A  
1. What is an Object and a Class?**

A Class is an encapsulation of properties and methods that are used to represent a real-time entity. It is a data structure that brings all the instances together in a single unit. An Object in an instance of a Class. Technically, it is just a block of memory allocated that can be stored in the form of Variables, Array or a Collection.

**2. What are the fundamental OOP concepts?**

The four fundamental concepts of Object-Oriented Programming are: Encapsulation: The Internal representation of an object is hidden from the view outside object’s definition. Only the required information can be accessed whereas the rest of the data implementation is hidden.   
Abstraction: It is a process of identifying the critical behavior and data of an object and eliminating the irrelevant details.   
Inheritance: It is the ability to create new classes from another class. It is done by accessing, modifying and extending the behavior of objects in the parent class.   
Polymorphism: The name means, one name, many forms. It is achieved by having multiple methods with the same name but different implementations.

**3. What is Managed and Unmanaged code?**

Managed code is a code which is executed by CLR (Common Language Runtime) i.e all application code based on .Net Platform. It is considered as managed because of the .Net framework which internally uses the garbage collector to clear up the unused memory. Unmanaged code is any code that is executed by application runtime of any other framework apart from .Net. The application runtime will take care of memory, security and other performance operations.

**4. What is an Interface?**

An Interface is a class with no implementation. The only thing that it contains is the declaration of methods, properties, and events.

**5. What are the different types of classes in C#?**

Partial class: Allows its members to be divided or shared with multiple .cs files. It is denoted by the keyword Partial.

Sealed class: It is a class which cannot be inherited. To access the members of a sealed class, we need to create the object of the class. It is denoted by the keyword Sealed.

Abstract class: It is a class whose object cannot be instantiated. The class can only be inherited. It should contain at least one method.   
It is denoted by the keyword abstract.

Static class: It is a class which does not allow inheritance. The members of the class are also static. It is denoted by the keyword static. This keyword tells the compiler to check for any accidental instances of the static class.

**6. Explain Code compilation in C#.**

There are four steps in code compilation which include:   
1.) Compiling the source code into Managed code by C# compiler.   
2.) Combining the newly created code into assemblies.   
3.) Loading the Common Language Runtime(CLR).   
4.) Executing the assembly by CLR.

**7. What are the differences between a Class and a Struct?**   
Class: Supports Inheritance, pass by reference (reference type), Members are private by default, Good for larger complex objects, Memory management and Garbage Collector available.   
Struct: Does not support Inheritance, Pass by Copy (value type), Members are public by default, Good for Small isolated models, cannot use Garbage collector and hence no Memory management.

**8. What is the difference between Virtual method and Abstract method?**   
A Virtual method must always have a default implementation. However, it can be overridden in the derived class, though not mandatory. It can be overridden using override keyword.

An Abstract method does not have an implementation. It resides in the abstract class. It is mandatory that the derived class implements the abstract method. An override keyword is not necessary here though it can be used.

**9. Explain Namespaces in C#.**   
They are used to organize large code projects. “System” is the most widely used namespace in C#. We can create our own namespace and use one namespace in another, which are called Nested Namespaces. They are denoted by the keyword “namespace”.

**10. What is “using” statement in C#?**“Using” Keyword denotes that the particular namespace is being used by the program. For Example, using System. Here System is a namespace. The class Console is defined under System. So we can use the console.writeline (“….”) or readline in our program.

**11. Explain Abstraction.**   
Abstraction is one of the OOP concepts. It is used to display only the essential features of the class and hides the unnecessary information.

**12. Explain Polymorphism.**   
Programmatically, polymorphism means same method but different implementations. It is of 2 types, Compile-time and Runtime. Compile time polymorphism is achieved by operator overloading. Runtime polymorphism is achieved by overriding. Inheritance and Virtual functions are used during Runtime Polymorphism.

**13. How is Exception Handling implemented in C#?**   
Exception handling is done using four keywords in C#:   
1) try: Contains a block of code for which an exception will be checked.   
2) catch: It is a program that catches an exception with the help of exception handler.   
3) finally: It is a block of code written to execute regardless whether an exception is caught or not.   
4) Throw: Throws an exception when a problem occurs.

**14. What are C# I/O Classes?**

C# has System.IO namespace, consisting of classes that are used to perform various operations on files like creating, deleting, opening, closing etc.

**15. What is a Destructor in C#?**

A Destructor is used to clean up the memory and free the resources. But in C# this is done by the garbage collector on its own. System.GC.Collect() is called internally for cleaning up. But sometimes it may be necessary to implement destructors manually.

**16. What is an Abstract Class?**

An Abstract class is a class which is denoted by abstract keyword and can be used only as a Base class. An Abstract class should always be inherited. An instance of the class itself cannot be created. If we do not want any program to create an object of a class, then such classes can be made abstract. Any method in the abstract class does not have implementations in the same class. But they must be implemented in the child class.

**17. What are Boxing and Unboxing?**

Boxing: Converting a value type to reference type is called Boxing. int Value1 = 10; object boxedValue = Value1;   
Unboxing: Explicit conversion of same reference type (created by boxing) back to value type is called Unboxing. int UnBoxing = int (boxedValue);

**18. What is the difference between Continue and Break Statement?** Break statement breaks the loop. It makes the control of the program to exit the loop. Continue statement makes the control of the program to exit only the current iteration. It does not break the loop.

**19. What is the difference between finally and finalize block?**

finally block is called after the execution of try and catch block. It is used for exception handling. Regardless of whether an exception is caught or not, this block of code will be executed. Usually, this block will have clean-up code.   
Finalize method is called just before garbage collection. It is used to perform clean up operations of Unmanaged code. It is automatically called when a given instance is not subsequently called.

**20. What is a Jagged Array?**

A Jagged array is an array whose elements are arrays. It is also called as the array of arrays. It can be either single or multiple dimensions. int[] jaggedArray = new int[4][];

**21. Name some properties of Array**

1. Length: Gets the total number of elements in an array.

2. IsFixedSize: Tells whether the array is fixed in size or not.   
3. IsReadOnly: Tells whether the array is read-only or not.

**22. What is an Array Class?**

An Array class is the base class for all arrays. It provides many properties and methods. It is present in the namespace System.

**23. What is an Escape Sequence?**

Name some String escape sequences in C#. An Escape sequence is denoted by a backslash (\). The backslash indicates that the character that follows it should be interpreted literally or it is a special character. An escape sequence is considered as a single character. String escape sequences are as follows: \n – Newline character \b – Backspace \\ – Backslash \’ – Single quote \’’ – Double Quote

**24. What are Regular expressions?**

Regular expression is a template to match a set of input. The pattern can consist of operators, constructs or character literals. Regex is used for string parsing and replacing the character string.

**25. What is Parsing?**

Parsing is converting a string into another data type.

**26. What is a Delegate?**

A Delegate is a variable that holds the reference to a method. Hence it is a function pointer of reference type. All Delegates are derived from System.Delegate namespace. Both Delegate and the method that it refers to can have the same signature.

**27. What are Events?**   
Events are user actions that generate notifications to the application to which it must respond. The user actions can be mouse movements, keypress and so on. Programmatically, a class that raises an event is called a publisher and a class which responds/receives the event is called a subscriber. An Event should have at least one subscriber else that event is never raised. Delegates are used to declare Events.

**28. What are the different types of Delegates?**

Single Delegate: A delegate which can call a single method. Multicast Delegate: A delegate which can call multiple methods. + and – operators are used to subscribe and unsubscribe respectively. Generic Delegate: It does not require an instance of delegate to be defined. It is of three types, Action, Funcs and Predicate.   
Action: In the above example of delegates and events, we can replace the definition of delegate and event using Action keyword. The Action delegate defines a method that can be called on arguments but does not return a result.

**29. What are Synchronous and Asynchronous operations?** Synchronization is a way to create a thread-safe code where only one thread can access the resource at any given time. Asynchronous call waits for the method to complete before continuing with the program flow. Synchronous programming badly affects the UI operations, when the user tries to perform time-consuming operations since only one thread will be used. In Asynchronous operation, the method call will immediately return so that the program can perform other operations while the called method completes its work in certain situations.

**30. What is Reflection in C#?**

Reflection is the ability of a code to access the metadata of the assembly during runtime. A program reflects upon itself and uses the metadata to inform the user or modify its behavior. Metadata refers to information about objects, methods. The namespace System.Reflection contains methods and classes that manage the information of all the loaded types and methods. It is mainly used for windows applications, for Example, to view the properties of a button in a windows form.

**31. Explain Get and Set Accessor properties?**

Get and Set are called Accessors. These are made use by Properties. A property provides a mechanism to read, write the value of a private field. For accessing that private field, these accessors are used. Get Property is used to return the value of a property Set Property accessor is used to set the value.

**32. What are Async and Await?**

Async and Await keywords are used to create asynchronous methods in C#. Asynchronous programming means that the process runs independently of main or other processes.

**33. What is a Deadlock?**

A Deadlock is a situation where a process is not able to complete its execution because two or more processes are waiting for each other to finish. This usually occurs in multi-threading. Here a Shared resource is being held by a process and another process is waiting for the first process to release it and the thread holding the locked item is waiting for another process to complete.

**34. What is a Race Condition?**

A Race condition occurs when two threads access the same resource and are trying to change it at the same time. The thread which will be able to access the resource first cannot be predicted.

**35. What is Thread Pooling?**

A Thread pool is a collection of threads. These threads can be used to perform tasks without disturbing the primary thread. Once the thread completes the task, the thread returns to the pool. System.Threading.ThreadPool namespace has classes which manage the threads in the pool and its operations.

**36. What is Serialization?**

Serialization is a process of converting a code to its binary format. Once it is converted to bytes, it can be easily stored and written to a disk or any such storage devices. Serializations are mainly useful when we do not want to lose the original form of the code and it can be retrieved anytime in the future.

**37. What is an XSD file?**

An XSD file stands for XML Schema Definition. It gives a structure for the XML file. It means it decides the elements that the XML should have and in what order and what properties should be present. Without an XSD file associated with XML, the XML can have any tags, any attributes, and any elements. Xsd.exe tool converts the files to XSD format. During Serialization of C# code, the classes are converted to XSD compliant format by xsd.exe.