# What are similarities between Class and structure?

* Both can have constructors, methods, properties, fields, constants, enumerations, events, and event handlers.
* Structures and classes can implement interface.
* Both of them can have constructors with and without parameter.
* Both can have delegates and events.

# What is the difference between Class and structure’s?

* Structures are value types and classes are reference types. So structures use stack and classes use heap.
* Structures members cannot be declared as protected, but class members can be. You cannot do inheritance in structures.
* Structures do not require constructors while classes require.
* Objects created from classes are terminated using Garbage collector. Structures are not destroyed using GC.

# What is “Sealed” modifier?

* Sealed modifer is used on class to pervent it from inherited from other classes
* Sealed modifer on method prevent the method from override from inherited classes.

# List out the differences between Array and ArrayList in C#?

* Arrays keeps data of same type whereas ArrayList can have data of various data types
* Arrays are of fixed length whereas Arraylist grows as values are added

# Why use “using” keyword?

* “using” keyword is used to invoke namespaces, wherein we can use the classes from various namespaces without using it’s fully qualified name
* “using” is used as code-block on class that implements IDisposable interface
* “using” is used as code-block wherein it calls the “dispose” method internally on objects used in using code block. However, if exception occurs on the using statement then it doesn’t get caught.

Ex: using( ){};

# Explain Namespaces in C#?

* Namespaces are containers to group related classes.

# Difference between Const, readonly and static keywords?

* Const: variable value has to be assigned at declaration. And it’s value cannot be changed later in the code. By default, const is static as it’s value cannot be changed once assigned. Constants can be accessed by using class name directy without creating instances because by default it is static, it cannot be accessed by using the instance of the class but inside the call without class name. Ex: const int fixedpoint = 21;
* Readonly: A ReadOnly variable is similar to constant but we can assign the value either at the time of declaration or in the constructor. values can be changed at runtime via (only constructors). Also, readonly variable can be re-assigned with new value at runtime (via parameterised constructor).
* Static readonly : variable value can be assigned at variable or static constructor. It’s value can be changed only once (via static constructor)
* Static : is a class level variable. Hence, only one copy is shared across all instances.

# Difference between Interface and Class

* Interface contains only member declaration (method,properties,indexes and Events), whereas Abstract class can have abstract and concrete methods.
* All members of Interface are public by default. Access modifiers are not allowed in Interface, whereas Abstract class can have various access modfiers (public,private,protected,protected internal)
* Interface facilitates mutiple inhertences. A class can implement multiple interfaces. Whereas Abstract class do not support multiple inheritence. A class can inherit fron only one base class
* Abstract class cannot be instantiated. Interface can.
* Interface cannot have fields and constructors, whereas Abstract class can have fields and constructors
* Abstract class should be used in situations wherein various related class have common members, the common members can be placed in an abstract class which then can be inherited by the classes. Abstract class also facititate to provide partial implementations for reuseability and productivity. Interfaces should be used in situations wherein various unrelated objects are required to implement certain functionalities. Inferface can also be used in situation when you cannot use class inheritance. For example, structures cannot inherit from classes, but they can implement interfaces.

# What is Enum?

* Enum is a set of named constants. It’s default type is int and starts with “0” for the element constant. Enumerations (enums) make your code much more readable and understandable.

# Difference between ref and out keyword?

* Ref parameter is used when a value type has to be passed as reference type in the callin function.
* Ref parameter value can be changed from within the calling method
* Ref parameter has to be initialized before passing into the calling method.
* Out parameter is used when multiple values has to be returned from the callling method.
* Out parameter has to be initialized/ assigned from within the calling method.

# What is property?

* Property are public data member to access private fields of a class.
* There can be 3 variasions : read/write, read, write.

# Difference between dispose and finalize?

* Finalize method is used to re-claim the memory from unmanaged resources (such as database connection etc). It cannot be called from user code rather it is called by Gargbage Collector. Finalize method cannot be implemented directly it can only be implemented via destructors. After compilation destuctors becomes finalize method.
* Dispose method is also used to re-claim the memory from unmanaged resources, however it can be explicity implement and invoked from user code. Idisposable interface should be implemented to use dispose method in user-defined class.

# Difference between String and StringBuilder?

* String: is immutable object means it takes new memory whenever a new value is assigned/append. It belongs to System namespace.
* StringBuilder: is mutable objects. It belongs to System.Text

# Partial Classes

* A class defination can split in multiple files with the same namespace. Partial keyword should be used to split class defination, and it is complied as single class.

# Difference between IEnumerable and IQueryable ?

* IEnumerable exists in System.Collection Namespace. VS . Iqueryable exists in System.Linq namespace
* IEnumerable suits to query in-memory data (such as collections). Vs IQueryable is used to query out-of-memory data (like database).
* IEnumerable do not supports Lazy Loading. VS. IQueryable supports Lazy support. Hence suitable to Paging like senario.
* Extention methods supports by IEnumerable takes functional objects. VS. Extension methods supports by IQueryable takes expression objects means expression tree.

# Difference between IEnumerable and IList ?

1. IEnumerable exists in System.Collections Namespace.
2. IEnumerable can move forward only over a collection, it can’t move backward and between the items.
3. IEnumerable is best to query data from in-memory collections like List, Array etc.
4. IEnumerable doesn't support add or remove items from the list.
5. Using IEnumerable we can find out the no of elements in the collection after iterating the collection.
6. IEnumerable supports deferred execution.
7. IEnumerable supports further filtering.
8. IList exists in System.Collections Namespace.
9. IList is used to access an element in a specific position/index in a list.
10. Like IEnumerable, IList is also best to query data from in-memory collections like List, Array etc.
11. IList is useful when you want to Add or remove items from the list.
12. IList can find out the no of elements in the collection without iterating the collection.
13. IList supports deferred execution.
14. IList doesn't support further filtering.