# 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. Immutable means: value cannot be changed,instead create new.
* 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.

# Difference between Single, SingleOrDefault, First and FirstOrDefault

* Single: return single specific element from a collection of elements if element match found. An exception is thrown, if none or more than one match found for that element in the collection
* SingleOrDefault : returns a single specific element from a collection of elements if element match found. An exception is thrown, if more than one match found for that element in the collection. A default value is returned, if no match is found for that element in the collection.
* First : returns first specific element from a collection of elements if one or more than one match found for that element. An exception is thrown, if no match is found for that element in the collection.
* FirstOrDefault : returns first specific element from a collection of elements if one or more than one match found for that element. A default value is returned, if no match is found for that element in the collectio

So, when to use appropriate above extention method:

1. When you want an exception to be thrown if the result set contains many records, use Single or SingleOrDefault.
2. When you want a default value is returned if the result set contains no record, use SingleOrDefault.
3. When you always want one record no matter what the result set contains, use First or FirstOrDefault.
4. When you want a default value if the result set contains no record, use FirstOrDefault.

# Difference between Select and SelectMany in LINQ

Select and SelectMany are projection operators. Select operator is used to select value from a collection and SelectMany operator is used to select values from a collection of collection i.e. nested collection.

**List<Employee> employees = new List<Employee>();**

**Employee emp1 = new Employee { Name = "Deepak", Skills = new List<string> { "C", "C++", "Java" } };**

**Employee emp2 = new Employee { Name = "Karan", Skills = new List<string> { "SQL Server", "C#", "ASP.NET" } };**

**IEnumerable<List<String>> resultSelect = employees.Select(e=> e.Skills);  
IEnumerable<string> resultSelectMany = employees.SelectMany(emp => emp.Skills);**

# Can we execute multiple catch blocks in C#?

No. Once any exception is occurred it executes specific exception catch block and the control comes out.

# **what is the difference between “throw ex” and “throw” methods in C#?**

* “throw ex” will replace the stack trace of the exception with stack trace info of re throw point.
* “throw” will preserve the original stack trace info.

# **Can we have only “try” block without “catch” block in C#?**

Yes.

# **Mention the assembly name where System namespace lies in C#?**

Mscorlib.dll

# **Can we use “this” inside a static method in C#?**

No. We can’t use “this” in static method

# **Can we override private virtual method in C#?**

No. We can’t override private virtual methods as it is not accessible outside the class.

# What is virtual Methods?

Virtual methods can be Overidden in derived class.   
Virtual method cannot be declared as static, or private

# **Explain access modifier – “protected internal” in C#?**

Can be used in same assembly and inherited classes.

# **In try block if we add return statement whether finally block is executed in C#?**

Yes.  
What is the difference between methods – “System.Array.Clone()” and “System.Array.CopyTo()” in C#?

* “CopyTo()” method can be used to copy the elements of one array to other.
* “Clone()” method is used to create a new array to contain all the elements which are in the original array.

# **How we can sort the array elements in descending order in C#?**

“Sort()” method is used with “Reverse()” to sort the array in descending order.

# **What you mean by delegate in C#?**

Delegates are type safe pointers. **delegate** is a reference type variable that holds the reference to a method (v. The reference can be changed at runtime. **Delegates** are especially used for implementing events and the call-back methods.

# **What are the types of delegates in C#?**

* Single Delegate
* Multicast Delegate
* Generic Delegate

# **What are the three types of Generic delegates in C#?**

* Func
* Action
* Predicate

# **What are the differences between events and delegates in C#?**

To be determined...

# **What are the uses of delegates in C#?**

* Callback Mechanism
* Asynchronous Processing
* Abstract and Encapsulate method
* Multicasting

# **What is Nullable Types in C#?**

Enables Value Types to hold “null” as an value. E.g: **int? X = null;**

# **Why to use “Nullable Coalescing Operator” (??) in C#?**

Nullable Coalescing Operator can be used with reference types and nullable value types. So if the first operand of the expression is null then the value of second operand is assigned to the variable. For Example:

double? myFirstno = null;  
double mySecno;  
mySecno = myFirstno ?? 10.11;

# **What is the difference between “as” and “is” operators in C#?**

* “as” operator is used for casting object to type or class.
* “is” operator is used for checking the object with type and this will return a Boolean value

# **Why to use lock statement in C#?**

Lock will make sure one thread will not intercept the other thread which is running the part of lock code. So lock statement will make the thread wait, block till the object is being released. In short, it synchonizes access to share code

# What is static constructors?

Static constructors are used to initialize static memebers of the class. It is invoke only once when the class is referenced for the first time.

# What is Monitor Class?

Monitor class provides a mechanism that synchronize access to an object. It ensure only one thread gets access to shared resource.

# Is overriding function allowed in same class?

No.

# What is System.Collections?

System.Collections namespace is from System (mscorlib.dll) core library. It has below collection objects:

# ArrayList

* BitArray
* HashTable
* Queue
* Stack
* SortedList

# What is ArrayList?

ArrayList is a dynamic array. Elements can be added / removed at runtime. Initially, the size of the arraylist remains to 16 elements, and grows with another 16 elements,with every seventeenth element added. It is not thread safe. ArrayList is not sorted by default. It is not generic

# What is BitArray?

Manages a compact array of bit values, which are represented as Booleans, where **true** indicates that the bit is on (1) and **false** indicates the bit is off (0).

# What is HashTable?

Represents a collection of key/value pairs that are organized based on the hash code of the key. It is not thread-safe.

# What is Queue?

Represents FIFO (First In First Out) data collection

# What is Stack?

Represents LIFO (Last In First Out) data collection

# What is sortedList?

Represents a collection of key/value pairs that are sorted by the keys and are accessible by key and by index

# What is Reflection?

Reflection helps to get meta-data information of objects at runtime.

# What is difference between System.Array.CopyTo() and Array.Clone()?

System.Array.clone() : This creates a new array and copies all the elements to it.

System.Array.CopyTo() : This copies the elements to an existing array.

Both perform shalow copy, means, the contents (each array element) contains references to the same object as the elements is the original array. A deep copy would create a new instance of each element’s objects,resulting a different yet identical array.

# What is underneath the sortedList class?

A sorted HashTable

# How to load Assembly from GAC?

Use:   
AssemblyName am = new AssemblyName(“”);

Assembly al = Assembly.Load(am);

# What is difference between DataReader and DataSet?

DataReader: is a connected,read-only and forward-only record set.

DataSet: is a disconnected in-memory representation of data.

# What is difference between tryParse and Parse?

TryParse : returns bool value stating the parse is successful or not.   
Parse : Parse an string value, and raise Invalid Cast Exception if value parsing fails.

If(Int32.tryParse(value, out val )){.. };

# What is difference between a==b and a.equals(b)?

For Value Type Data: both works same. Compares the values  
For Reference Type Data : both works differently.   
 a==b : compares the references. Means, if a and b are both pointing to the same object.  
a.equals(b) : it compares the values of Objects. If a and b objects have equivalent values.

Recommendations:   
For value Type : use “==” . For Reference Type : use equals()

# What is optional and Named Parameter?

**Optional Parameter** : allows a function to be created specifying default values. So that when the function is invoked optional parameters can be ommited. Optional parameters are always declared as the last one in parameter sequence.

Function callme( int a, int b =2, int c=4) { .. } // Declaration

Callme(10) // Invoke method specifiying only required one

**Named Parameter :**  allows a method to be called bt specifiying the parameter Name and value. Like below:   
 Callme(10,c:5); // is equvalent to Callme(10,2,5);

# What is Anonymous Method?

C# 3.0 : allows to create method which has a body but do not specify a method Name. It gets executed only from the same code location and cannot be invoked excipilitly.

# What is difference between var and dynamic keyword?

Var : keyword is introduced in C# 3.5 with LINQ. It is a keyword which actual type is inferred by compiler

Dynamic: keyword is introduced in C# 4.0. It is evaluated only at runtime. So,any error/exception gets caught at runtime.

# What is C# Preprocessor?

Preprocessor : are special directives that gets processed before compiler starts processing code.

#region : #endregion -: used to collapse a section of code.

#define : #undef -: Used to define , undifine conditionall symbols

#if : #elIf : #else : #endif -: Used to conditionally compile or uncompile code

# Which classes are required to executed javascript for code behind?

RegisterStartupScript: adds script at the beginning at the <script> tag

RegisterClientScriptBlock : adds script at end of the <script> tag

# What is Polymorphism? Runtime Polymorphism and Compile Time polymorphism

Polymorphism is the ability of code module to behave differently based on input message.

Runtime Polymorphism : Method Overridding

Compile Polymorphism : Method Overloading

# Do Constructor and Destructor are inherited to derived class?

No.

# Partial Methods in C#?

Partial methods are declared and defined on in Partial class or partial struct.  
Partial methods are only of private access modifier, with void return type.  
Partial method is two step activity. First, Need to be declared and then it’s body can be defined in the same partial class or another partial class.  
Partial methods cannot use “out” keyword

# Difference between Debug.write and Trace.write?

Debug.write : works on Debug mode and release mode  
Trace.write : works in Release mode only.

# What is Structs?

Struct is similar to class, however it is System.ValueType. Struct can have members such as constructor, methods and finalizer.

# What is Tuple?

A tuple is a data structure that provides an easy way to represent a single set of data.

e.g: Tuple<int,int,int> t1 = new Tuple<int,int,int>();

# What is Co-variance and Contra-variance in C# 4.0?

Converting a broader type to a lower herarchy specific type is Contra-variance. And Converting from a specific derived type to broader type is **covariance**.   
In C# 4.0 : Covariance allow to convert an collection of objects from derived type to broader type. And Contra-variance allows to convert an collection of objects from broader type to derived type.

e.g:   
List<Animal> al = new List<Animal>();   
List<Cat> ct = al; // covariance. Implict conversion

List<Cat> ct = new List<Cat>();  
List<Animal> al = ct ; //Contra –variance. Data

# What is shallow Copy and Deep Copy?

# Namespace for Regular Expression?

System.Text.RegularExpression;