**What is Array List in C#?**

In C#, the Array List is **a non-generic collection of objects whose size increases dynamically**. It is the same as Array except that its size increases dynamically. An Array List can be used to add unknown data where you don't know the types and the size of the data.

**What is difference between Array and List?**

Array stores fixed size sequential collection of elements of the same type where as List is a generic collection.

Array can directly handle arithmetic operations where List cannot.

**What is Multi-Dimensional array and how to declare it?**

Int[,] ar = new int[2,3] // array of 2 rows and 3 columns

Console.Writeline (ar[0,0] + “ ” ar[0,1] + “ ” ar[0,2]);

Console.Writeline (ar[1,0] + “ ” ar[1,1] + “ ” ar[1,2]);

**What is difference between Array and ArrayList?**

|  |  |
| --- | --- |
| Array | ArrayList |
| We cant revise length of the array once declared | We can revise the length |
| Array can hold primitive as well as object data types | ArrayList can hold only object data types |
| Single or multi dimentional | Only single dimentional |
| Faster due to static behaviour | Slower due to dynamic behaviour |

What is Struct?

Structure is an user defined data type which stores collection of variables of different types

What is difference between Struct and Class?

Structures are value types where class is reference type

Structure elements are public by default where class members are private by default

What is difference between Abstract class and interface?

|  |  |
| --- | --- |
| Abstract Class | Interface |
| It contains both declaration and definition part. | It contains only a declaration part. |
| Every method present in Abstract class need not be public and abstract. In addition to abstract methods we can take concrete methods also | Inside Interface every method is always public and abstract whether we are declaring or not. Hence Interface is also considered as 100% pure abstract class. |
| Multiple inheritance is not achieved by abstract class. | Multiple inheritance is achieved by interface. |
| It contain [constructor](https://www.geeksforgeeks.org/c-sharp-constructors/). | It does not contain [constructor](https://www.geeksforgeeks.org/c-sharp-constructors/). |
| It can contain static members. | It does not contain static members. |
| It can contain different types of access modifiers like public, private, protected etc. | Interface cannot have access modifier TO its members as they are public by default |
| The performance of an abstract class is fast. | The performance of interface is slow because it requires time to search actual method in the corresponding class. |
| It is used to implement the core identity of class. | It is used to implement peripheral abilities of class. |
| A class can only use one abstract class. | A class can use multiple interface. |
| If many implementations are of the same kind and use common behavior, then it is superior to use abstract class. | If many implementations only share methods, then it is superior to use Interface. |
| Abstract class can contain methods, fields, constants, etc. | Interface can only contain methods and does not contain fields like int a, string str |
| It can be fully, partially or not implemented. | It should be fully implemented. |
| If we are talking about implementation but not completely (partial implementation) then we should go for abstract class. | When we don’t now anything about implementation just we have requirement specification then we should go for interface. |

We cannot create an object of an abstract class.

If you want to use it then it must be inherited in a subclass.

Interfaces allow us to develop loosely coupled systems

Interfaces are very useful for dependency injection

Interfaces make unit testing and mocking easier

You should use an interface if you want a contract on some behavior or functionality. You should not use an interface if you need to write the same code for the interface methods. In this case, you should use an abstract class, define the method once, and reuse it as needed

**What is OUT parameter and ref parameter?**

Out parameter is used to return the output. We can declare and return more than one out parameters.

Ref parameter is used to pass actual memory location of the object

**What is Collection in C#?**

Collection is nothing but group of records which can be treated as one logical unit.

Eg: List, Arrays, Stacks, Queues

**What is Static Class?**

A static class is declared with the help of *static*keyword. A static class can only contain static data members, static methods, and a static constructor. It is not allowed to create objects of the static class. Static classes are [***sealed***](https://www.geeksforgeeks.org/c-sealed-class/), means one cannot inherit a static class from another class.

Static classes are used as the container for static members like methods, constructors and others.

**What is Static Variable?**

A static variable is declared with the help of static keyword. When a variable is declared as static, then a single copy of the variable is created and shared among all objects at the class level. Static variables are accessed with the name of the class, they do not require any object for access.

**What is mean by Const Keyword?**

a variable whose value will not change during the lifetime of the program.

**What is Lazy Loading and Eager Loading?**

Lazy loading is a concept where we delay the loading of objects and load them only when they are required

Eg – Employees having tickets associated with them but on Employee list page we not need to load tickets also instead of we can load them only on click of an Employee(On Employee details page)

**What is difference between Constant and Readonly?**

Constant is a compile time constant while readonly is the runtime constant. Means we have to initialize constants value at the time of declaration while we can initialize readonly after declaring it.

**What is difference between CHAR and VARCHAR?**

CHAR is of fixed length while varchar is of variable length like Array and ArrayList. If we declare like this:

CHAR(15);

VARCHAR (15);

And we are inserting ABC in both, so CHAR will still take 15 bytes to store ABC while VARCHAR will take only 3 bytes.

**What is ASCII and UNICODE?**

In computer to transfer the data we need to convert the characters into 0 and 1 format so for this conversion we firstly have a technique is ASCII(American Standard Code for Information Interchange)

ASCII has two types:

ASCII – 7 – can convert 128 types of characters

ASCII – 8 – can convert 256 types of characters.

But in world we have many types of languages and many symbols and letters for that languages so 256 types was not enough to convert all these languages. So to overcome this problem UNICODE introduced

In UNICODE we have 3 types that are:

UTF -8 (Unicode Transformation Format 8bit)

UTF -16 (Unicode Transformation Format 8bit)

UTF -32 (Unicode Transformation Format 8bit)

**What is the difference between varchar and nvarchar?**

NVARCHAR stores UNICODE data while VARCHAR stores ASCII data. NVARCHAR uses 2 bytes per character while VARCHAR uses 1 bytes per character.

**What is difference between String and string?**

Essentially, there is no difference between string and String (capital S) in C#.

String (capital S) is a class in the .NET framework in the System namespace. The fully qualified name is System.String. Whereas, the lower case string is an alias of System.String.

**What is Parent class of string and object?**

Parent class of string is System.String and parent class of System.String is Objects

**What is State Management?**

In web applications when we move from one page to another page the data from first page disappears as the web applications are stateless. To maintain this data we need to do State Management.

State Management we can achieve by two methods:

**Client Side**:

In Client Side we have following techniques:

By using cookies

By using hidden fields

By using view states

By using query string

**Server Side:**

In Server Side we have following techniques:

Below two techniques we used while we have authentication and authorization implemented in our application, In this case we store logged in user credentials of the user on the server in one file so that we can make use of them and user not needed to provide credentials on each page

**Sesssion State** : In session state separate file for separate user gets created on the server for storing credentials

**Application State** : In application state one single file gets created for all the users for storing credentials.

**What is Extension Method in C#?**

A C# extension methods allows developers to extend functionality of an existing type without creating a new derived type, recompiling, or otherwise modifying the original type.

namespace ExtensionMethod1

{

public static class XX

{

public static void NewMethod(this Class1 ob)

{

Console.WriteLine("Hello I m extended method");

}

}

class Program

{

static void Main(string[] args)

{

Class1 ob = new Class1();

ob.Display();

ob.Print();

ob.NewMethod();

Console.ReadKey();

}

}

}

In the above code, you see there is a static class XX with a method, NewMethod. If you notice, the NewMethod takes Class1 as a parameter. This is how you extend an existing library and add your own methods to it. Now, this New Method is available via the class library.

Eg 2

If we have a String class and in String we can have Substring method available but if we want the substring of last 5 letters then we neeed to add this method as an extension method in String class

Public Static class StringExtensions

{

Public static string RightSubstring(this String s, int count)

{

Return s.Substring(s.Length-count,count);

}

}

**Access Modifiers in C#**

**Public** – The code is accessible for all classes

**Private** – The code is accessible within the class

**Protected** – The code is accessible within the class and in inherited classes

**Internal** – The code is accessible within the assembly/same project

**Protected Internal** - Access is limited to the current assembly or types derived from the containing class.

**Private Protected** - Access is limited to the containing class or types derived from the containing class within the current assembly.

**What is default access modifier in c#?**

Default access modifier of class is Internal. And private for class member.

**What is Partial Classes**

It is a type of class that allows dividing their properties, methods and events into multiple source files and at compile time these files are combined into a single class.

The following are some key points:

• All the parts of the partial class must be prefixed with the partial keyword.

• If you seal a specific part of a partial class then the entire class is sealed, the same as for an abstract class.

• Inheritance cannot be applied on partial classes.

• The classes that are written in two class files are combined together at run time.

**Partial class in different namespaces possible?**

A class's name includes its namespace, so name1.Foo and name1.name2. Foo are two completely separate types. So, the short answer to your question is: No.

**What is Sealed Class**

A Sealed class is a class that cannot be inherited and used to restrict the properties.

The following are some key points:

• A Sealed class is created using the sealed keyword.

• Access modifiers are not applied to a sealed class.

• To access the sealed members, we must create an object of the class.

**What Differences between IEnumerable and IQueryable?**

**IEnumerable**

1. IEnumerable exists in the System.Collections namespace.

2. IEnumerable is suitable for querying data from in-memory collections like List, Array and so on.

3. While querying data from the database, IEnumerable executes "select query" on the server-side, loads data in-memory on the client-side and **then filters the data**.

4. IEnumerable is beneficial for LINQ to Object and LINQ to XML queries.

IEnumerable takes all data from server to client side memory first and then filters it

**IQueryable**

1. IQueryable exists in the System.Linq Namespace.

2. IQueryable is suitable for querying data from out-memory (like remote database, service) collections.

3. While querying data from a database, IQueryable executes a "select query" on server-side with all filters.

4. IQueryable is beneficial for LINQ to SQL queries.

IQueryble filters the data on server only instead of loading all data to client memory

**What is serialization in c#?**

Serialization is the process of converting object into byte stream. The reverse process of serialization is called deserialization.

Serialization is internally used in remote applications. Serialization is process used to transform object over network or out of your process boundary.

**What is async and await in C#?**

Nowadays, Asynchronous programming is very popular with the help of the async and await keywords in C#. When we are dealing with UI, and on button click, we use a long-running method like reading a large file or something else which will take a long time, in that case, the entire application must wait to complete the whole task. In other words, if any process is blocked in a synchronous application, the whole application gets blocked, and our application stops responding until the whole task completes.

Asynchronous programming is very helpful in this condition. By using Asynchronous programming, the Application can continue with the other work that does not depend on the completion of the entire task.

**What is Private Constructor?**

A private constructor is a special instance constructor. It is generally used in classes that contain static members only. If a class has one or more private constructors and no public constructors, other classes (except nested classes) cannot create instances of this class.

Private constructors are used to prevent creating instances of a class

Use private constructor when class have only static members.

If we don’t use any access modifier to define a constructor, then the compiler takes that constructor as a private.

**What is Static constructor?**

A static constructor is used to initialize any [static](https://docs.microsoft.com/en-us/dotnet/csharp/language-reference/keywords/static) data, or to perform a particular action that needs to be performed only once. It is called automatically before the first instance is created or any static members are referenced.

**Difference between Private Constructor and Static Constructor**

Static Constructor

1. Used to initialize the static members of a class.
2. Can not access non-static members.
3. Executes before the first instance of a class. We can not determine the time of execution.
4. Executes by the CLR not by the object of a class.
5. There are no parameterized static constructors since it is handled by the CLR not by the object.
6. Time of execution might be at the loading of contained assembly.

Private Constructor

1. Used to restrict a class to be instantiated and to be inherited.
2. Used whenever a class contains only static members.

1)A static constructor is called before the first instance is created. i.e. global initializer.

Whereas Private constructor is called after the instance of the class is created.

2)Static constructor will be called first time when the class is referenced. Static constructor is used to initialize static members of the class.

**What is Finalize and Dispose methods in C#?**

Methods dispose() and finalize() are the methods of C# which are invoked to free the unmanaged resources held by an object. The dispose() method is defined inside the interface IDisposable whereas, the method finalize() is defined inside the class object. The main difference between dispose() and finalize() is that the method **dispose**() has to be explicitly invoked by the user whereas, the method **finalize()** is invoked by the garbage collector, just before the object is destroyed.

| **BASIS FOR COMPARISON** | **DISPOSE( )** | **FINALIZE( )** |
| --- | --- | --- |
| Defined | The method dispose( ) is defined in the interface IDisposable interface. | The method finalize( ) id defined in java.lang.object class. |
| Syntax | public void Dispose( ){ // Dispose code here } | protected void finalize( ){ // finalization code here } |
| Invoked | The method dispose( ) is invoked by the user. | The method finalize( ) is invoked by the garbage collector. |
| Purpose | Method dispose( ) is used to free unmanaged resources whenever it is invoked. | Method finalize( ) is used to free unmanaged resources before the object is destroyed. |
| Implementation | The method dispose( ) is to be implemented whenever there is a close( ) method. | The method finalize( ) is to be implemented for unmanaged resources. |
| Access specifier | The method dispose( ) is declared as public. | The method finalize( ) is declared as private. |
| Action | The method dispose( ) is faster and instantly disposes an object. | The method finalize is slower as compared to dispose |
| Performance | The method disposes( ) performs the instantaneous action hence, does not effect the performance of websites. | The method finalize( ) being slower affects the performance of the websites. |

**What are the SOLID principles?**

S – Single responsibility Principle – One class should have one single responsibility.

O – Open Close Principle – Methods should open for extensions but closed for modifications.

L – Liskov Substitute Principle – Any class must be directly replacable by any of its subclass

I – Interface Segregation Principle- Interfaces should be small and lightweight

D – Dependancy Inversion Principle – Superclasses should not be dependant on subclasses

**What is meaning of Using keyword in C#?**

Using keyword is used to dispose the object after use

**What is the difference between string and stringBuilder?**

String is immutable type where StringBuilder is mutable type. Immutable means each time when we modify the string the new instance will get created, where in mutable same instance will get modified

**What is difference between .equals and ==?**

== compares object references

.equals () compares object contents

**What is the difference between Let and Var keywords?**

Both let and var are used to declare the variables just scope of let is within the block it is declared and the scope of var is within the function it is declared.

**What is difference between var and Dynamic?**

var at the compile time(like constant) while dynamic are at run time(like readonly)

what

**What is the use of params keyword in C#?**

When developer does not know the number of parameters to pass or when we have to pass the list of parameters then we can use params keyword.

**What is use of this keyword?**

this keyword is used to refer to the current instance of the class.

This keyword avoids the name confusion between class fields and constructor parameters.

**What is Method Hiding?**

In method hiding we can hide the implementation of the methods of a base class from the derived class using the new keyword.

We can have completely new implementation for the methods in the base class into the derived class

**What is Generics in C#?**

If we have one method which accepts string parameters and we are calling that method by passing the int parameter then it will give error like “cannot convert type int to string” in this case if we make that method as generic by using <T> then it will accept any type of parameters. This is Generics.

Public static bool AreEqual <T> (T value1, T value2)

{

Return value1.Equals(value2);

}