**Q 1) What are the Advantage of Oops.**

1.Reuse of code by Inheritance 2) Flexibilty and more redability by polymorphism

3)Hiding Implementation by Abstraction 4)Hiding Data using Encapsulation binding data and behaviour into a single unit.

**Q-2) What are limitation of oops**

It is not suitable for small application. it is suitable for large application .

**Q-3)What are types of Inheritacne:-**

**1)Single Inheritance**:- one base class and one derived class

2) Multiple Inheritance :- Multiple Inheritance can be achieved by in c# by an Interface .

3)MultiLevel Inheritance :- in this type of Inheritance we will have mutilple base interface or class and mutilple child class

4)Hierarchical Inheritance :- means one base class but multiple child class like A is Base class B,,C and D are extending A .

**Q-4) How to Prevent a class from being Inherited**

By using sealed keyword with class

**Q-5) type of polymorphism:-**

1. Compile Type Polymorphism :- why it is called CompleTime Polymorphism

Because compiler Know already that these two methods are different . there fore which method is going to brecalled it is binded at compile time.

1. Run time Polymorphism:-

Overriding method have same name and same type of argument . so at compile time compiler does not know that particular refrence is pointing to or has which type of Instance.

**Q-6) when to use Abstract Class and when Interface.**

1. Abstract class is a good choice when you are sure some methods are should me implemented in a same way in all derived class and also when u have any instance variable that u are sure that this instance variable must be in all derived class. Forexample Balance , Balance is a variable that must be in Account class all derived class.
2. Interface is used to define only set of services. U know that the classes which implements this interface that must contains methods but it can be implementation different.

**Q-7) ref and out what is that , and what is different between them.**

If we pass any primitive type variable to a function that then copy of variable is created and stored that variable at different memory location. Because of that changed value will not occurs in variable which is passed. As we know that we can return only single value by function. So if we need return multiple value from function we use ref and out . if we use ref and out basically we don’t pass value we pass address , so changes will happen in paased variable.

Ref parameter variable must be initialized befor passing it . but it does not require in out parameter.

Second difference is the function in which we are passing is must to initialized out parameter before the completion of

Function.

**Q-8) What is the purpose of Params Keyword in c#.**

Params keyword is used when we require n number of parameter.

**Q-9) what are the Access Specifier ? what is default access specifier in a class.**

Access Specifier is the keyword which is used to define visibility of class , variables and methods

In c# there are 5 access Specifier

A ) Private :- the member which is define private access specifier is visible only inside the class.it is used only member of class.

B) public :-it is accessible everywhere means outside the project or assembly as well as inside assembly

C) protected :- it is accessible only in derived class in outside the assembly as well as inside assembly.

D)Internal :- when we define member with Internal access specifier it would be accessible only inside the assembly

E) Internal Protected :- inside assembly and outside of assembly only in derived class.

*Note:- default access specifier of class is private*

**Q 10) Exception Handling:-**

Exception handling in object oriented programming is used to manage the Errors.

The keyword which is used to handle exception are try , catch , finally

1. Try:- inside try block we write those code where there is chance of to occurs the exception.
2. Catch :- just after try block we use catch block, that is used to catch the exception.in catch block we write those code which will print which type of Exception occurred during the running of application.
3. Finally. Finally block basically it is used to write those code which would relase the resources. Bcz clr gives assurance to execute the finally block.

**Q 11) can we execute multiple catch block.**

No we cant execute multiple catch block. But we can write it .but one of them would execute.

Note:- only try will not allow but try with finlly is allowed

**Q 12) What is difference between “throw ex” and “throw”**

Throw key word gives more stack trace details as compare to throw ex

**Q 13) What is difference between ArraList and HashTable**

ArraysList we used to store the element. In this only element is stored. It works on dynamic resize able array.

On the other hand HashTable allows to store the element in key and value form. It is fast in searching as well as insertion and deletion.

**Q 14) Using:- using to import directive**

Using statement ensures that dispose() is called even if exception occurs.

The class object which we creating inside using block it must implement IDisposable Interface.so if we are opening session object inside using block that would be destroy by clr implicity call the dispose method.

**Q 15) what is diffrenece between is and as operator in c#**

Is keyword is used to check type of an object

Is return type is Boolean basically it is used for checking that particular refrence variable is pointing to any type on instance object or not. Like I have Account abstract class i have to implementation class that is Saving and current if I write Like that Account is Saving , so it would return true or false

As operator is used to perform convertion after it is compatible

But as key word is also used in same purpose but it return same object or null

**Q 16 ) what is difference between readonly and constant variable**

Readonly :- this keyword variable ,there can be assign a value while decleretion or in constructor

Read only is a runtime constant.

Constant:- but in case of constant variable while decleration variable must be initialized. Constant is compile time constant.

**Q 17 ) Boxing and Un Boxing**

int num=15; Object obj=num; this is boxing , boxing is conversion of value type to refrence type

boxing happens implicitly

2) num=(int)obj this is an example of un boxing , Un boxing is converstion of refrence type to value type.

Unboxing happens Explicitly.

**Q 18) What is difference between String and String Builder ? when we should use it.**

In c# String is a refrence type. It is presents in System name space. It is character of sequence.

String is immutable object means once the String object is created it content can not be changed but we are explicity try to change it a new object is created with new String or of resultant String .

So if in our application once an object is created and its value again agin frequently changes then we should for string builder

Or else we should use String object.

**Q 19) what are Nullable types.**

We can not assign null to value type like int j = null but we can assign by using Nullable keyword or ? QMark

**Q 20) What are the important component of .Net framework ? what are their roles?**

CLR:- Common language Runtime (CLR) manages the execution of programs written in any lauguage that uses the .Net framework for example c#,F#,Vb.Net and so on. CLR does various operation like memory management , security checks etc.

**Q 21) What is an Assembly ?**

Assembly is unit of deployment like exe or dll

When we create a code and build the solution then .Net framework convert it into assembly which we can see inside bin folder.

There is 3 type of assemblies:

Private assemblies:- 56:06

**Q 22) Garbage Collector**

Garbage collector is a Thread which run behind of clr .it is responsible to release object from heap meamory which is created by new keyword. When an object would be unreach able then garbage collector would delete it from memory.

**Q 23) Can we force Garbage collector to run**

Yes by using GC.Collect() but this is not recommended instead use Dispose method.

**Q 24) what is difference between Thread and Process**

Thread is basic unit of a program. It is independent task which belong to same program.

When a program becomes in execution is called process. process is divided into number of no of process is called paging by operating system. We can say that process is collection of thread

**Q 25) what is reflection:-**

In c# reflection is a set of interfaces and class . by the help of these class and interface we can get meta data of an aseembly or object at run time.

**Q 26) What is Serialization in java?**

Serialization is a process of converting C# object to stream of byte. So that stream of byte we can send Over the network. Or database or in a file.

Why we convert it into Stream of Byte.

First of all it is platform independent .

It helps to preserving the state of object. and De serialzation requires less time to create an object than actual object created by new keyword. So it helps also to save time.

**String:-** The String class is defined in the [**.NET**](https://www.geeksforgeeks.org/c-net-framework-basic-architecture-component-stack/) base class library. **System.String class is immutable**, i.e once created its state cannot be altered.

**Differences between *String* and *System.String* :**  In C#, there is no fundamental difference between **String** and **System.String** because **String** is an alias for **System.String**. They refer to the same class in the .NET Framework. It's just a matter of preference which one you choose to use in your code.When you write **String** in your code, the C# compiler automatically resolves it to **System.String** at compile-time. Similarly, when you use **System.String**, it refers to the **String** class.

String str1 = "Hello"; System.String str2 = "Hello";

Both **str1** and **str2** are of the **System.String** type.

The reason for having an alias like **String** is to provide a shorter and more readable alternative when working with strings in C#. It helps to improve code readability and reduce verbosity.

**How C# code is run:-**

In C#, a program is run by executing the compiled executable file produced by the C# compiler (typically with a .exe extension). The execution process involves several steps:

1. Source Code Compilation: The C# source code files (.cs files) are compiled using a C# compiler (such as the Roslyn compiler). The compiler translates the human-readable C# code into an intermediate language called Common Intermediate Language (CIL) or Microsoft Intermediate Language (MSIL).
2. Intermediate Language Generation: The compiler generates one or more files containing the compiled intermediate language code (with a .dll or .exe extension). These files contain the necessary instructions and metadata to run the program.
3. Just-In-Time (JIT) Compilation: At runtime, the Common Language Runtime (CLR) in the .NET framework takes the compiled intermediate language code and performs a Just-In-Time (JIT) compilation. The JIT compiler translates the intermediate language code into machine code specific to the target platform.
4. Execution: Once the JIT compilation is complete, the CLR executes the compiled machine code. The CLR manages various aspects of program execution, including memory management, exception handling, and security.
5. Entry Point: The CLR looks for the entry point of the program, which is typically the **Main** method. It locates the class containing the **Main** method and invokes it to start the execution of the program.
6. Program Execution: The program's execution proceeds according to the statements and logic written in the **Main** method and any other methods and classes called from it. The program performs the desired tasks, such as displaying output, interacting with users, processing data, etc.
7. Termination: The program continues to execute until it reaches the end of the **Main** method or encounters an explicit termination statement like **return** or **Environment.Exit**. Once the execution completes, the program terminates, and any allocated resources are released.

It's important to note that the specific execution process may vary depending on the environment and runtime being used, such as the .NET Framework or .NET Core, and any additional frameworks or libraries involved. However, the general steps outlined above provide a high-level overview of how a C# program is run.

**Indexer:-** Indexer is a special type of property that allows a class and structure member data to be access like an array. When we define an indexer inside a class or structure internally it behaves like virtual array.

An indexer can be define as same way as property with this keyword and squre bracket. Modifier can be private, public , protected, internal.

**Generic:-** generic is type safe. Basically it is used to define for use our method and class for all data type.

It is used to define our method and class can be reusable for every data type.

Generic is as an advantage in performance wise because it removes the possibility of boxing and unboxing happen between value type to reference type .

In Java, generic type parameters like **T** are replaced at compile time.

Collection in C#:-

First of all collection is set of interfaces and class that is used to store and manipulate the object in an organized manner.

ICollection is the interface and there are many class which is directly and in directly implementing collection interface. Those clases is used for store the object. It is used to do the operation when we have collection of object.

Basically all the collection name space classes internally implementing different different data structure like array , linked list, stack, hash table , tree

In c# we have collection related three name space first is System . Collection second id System. Collections . Generic and third is System . Collections . Concurrent

**Generic Collection:-**

Generic collection in C# defined in system . Collection .Generic name space . it provides generic implementation of slandered data structure like linked list , stack , queue and dictionary. These class object are type safe. We must to define which type of object we are going to store in the collection object.

In this collection we have root interface that is I Collection which is extending I Enumerable Interface which support iteration for Collection object.

I Enumerable (it provides basic support for itrration)

I Collection ( it provides for basic support for addition and remove element)

IList( indexing) I set(Uniqueness) I Dictionary

List <V> Hash Set <V> Dictionary <k , V>

Sorted Set <V> Sorted Dictionary <K,V>

Sorted List< k, V>

List <V> :- List is a class which impalement I List interface. It internally implement Dynamic re size able array.

It allows duplicate element. It support indexing to search the element.

Hash set :- this class is used when we require uniqueness . we have to store unique object. It does not maintain insertion order. It uses hashcode and equals method to determined that object is identical or not

Sorted set:- it is store the element and maintain sorting of the element. It by default keep sorting of the element in acending order using I comparable or I Comparator. Sorted set uses BST for sorting. It takes more memory because in BT we know that one node refer to another node.but it provides fast operation. To find the element in sorted set wheter the element is present or not it takes log n time complexity. But in Hash set is on

Dictionary :- it allows the object to store element in key and value form . key must be unique and can not be null. Value can be null and duplicate

Q) C# collection is used for store the object , manipulate the data inside the stored collection. For example like we have numbers of employee object and we have to add in salary bonus on the some percentage of salary , these type of operation can be done using collection. Wehave three type of collection in c# .

1) System . Collections ArrayList , Hashtable, Queue , Stack

2) System .Collections . generic above metioned

3) System. Collection . Concurrent ConcurrentDictionary< key , value> , ConcurrentStack <T> , etc

Q) what is difference between array and arralist

1) the first difference is array and arraylist is array is a fixed type data structure , while initialization of an array we must to iniatilze the size of an array while in arraylist we don’t need to inialize the size of an array

2) array store the element in contiguous manner .while array list convert it into boxing and it store the object type.

3) array stores single type of element while arraylist store multiple type of object.

Q) what is difference between ArrayList and Hashtable

ArratList store the object in value only while Hashtable store the object in key and value form.

Object is accessed in ArrayList by index number , while in Hashtable object is accessed by key.

Delegate:- Delegate are the reference type. It refer to function. Delegate method signature should be same as whatever function we are going to hold .means the faction which we are storing to a delegate that delegate signature and function signature must match and also return type also match.

A delegate can be define outside the class as well as it can be define inside the class. But it is recommended it should be outside the class.

Delegate has no method body. It is type safe , object oriented and secure.

Multi Cast Delegate :- when a delegate refer to multiple method then it is called Multi cast delegate. When we invoke single time it will execute all the method. Basically multi cast delegate is made those delegate which has not return type

Steps while working with delegate are

1. Declare the delegate
2. Ste the target method
3. Invoke the delegate

In c# there are some predefine delegates are like func , Action , predicate

Func :- Func is generic delegate which is present in system . name space . it has zero or more input parameter and one out parameter. We can keep the refrence of method which has around 16 parameter. Means that there is 17 overloded of func delegate.

Action :- Action is generic delegate which is present in system . name space . Action delegate has 0 or more input parameter and not any out put parameter.

Predicate:- it is also generic and system. Name speace . it must take one parameter and return bool

Anonymous Method:- as the name suggest an Anonymous Method is a method without name . Anonymous Method can be define in c# using delegate keyword . and can be assign to a delegate type. Anonymous Method can be passed as a parameter

Lamda Expression :- Lamda Expression is a better way to represent Anonymous Method . over here we are not need to write delegate. We use arrow . we assign lamda expression to a delegate type. In lamda expression we are not need to define parameter type and also return type. Compiler it also it self write after seeing to delegate type and also return type.

**DataType:-** collection of class , interface, delegates and enums.

Console is a class which is present is system namespace. These libraries is a part of .Net framework.

Data Types: SByte -128 to 127 , Byte 0 to 255, Short , UShort,int , uint , long , ulong,

float is present system.single 4byte 7preceison (including before point after point)

double is present system.Double 8 byte 15-16 digits

decimal is present System . Decimal can be stored float and double value but digit allowed 28 to 29

float f= 12.254f

decimal d=12547889.2588m

char b , string

Escap Sequence \n ,\t, \’ ,\”,\\, ”D:\\adil\\Charp\\tutorial” less readable

@(verbatrim literal ) ignore escap sequence @”D:\adil\Charp\tutorial” more readable

**NameSpace:-** Name space is concept of grouping the class , interface and enum. Bcz of that we will not get ambiguity problem.

in a simple C# program, we use System.Console where System is the namespace and Console is the class. To access the class of a namespace, we need to use namespacename.classname. We can use **using** keyword so that we don't have to use complete name all the time.

ASP.Net MVC

**Q 27) What is MVC Architechture**

MVC stands for model View and controller. It is an architecture for creating web application. This architecture makes application loosely Coupled and advantage of this is many different programmer can work simultaneously work in each layer.

Model:- in Model layer we make pojo or poko class that has only private field and getter and setter method. it is application data layer.

Controller:- Controller layer act as an interface between Model and View component . in controller layer we write business logic .controller layer contact to database and it would fetch the data from database and update ,delete etc.

View :- data sent by Controller layer is consumed in View Layer. View act as user Interface.

**Q 28) what are the different return type of Controller action method.**

1) ViewResult

2) Json Result

3) File Result

4) ContentResult

Q 29) what are filters and their types in MVC

Asp.Net MVC filter is a custom class where you can write custom logic to execute before or after an action method execution.