C# Interview Q & A

Palle Technologies

What is a class?

Class is a virtual entity or a model or a blue print.

Class is used for storing related variables & methods.

```
public  class  Student
{
      public  string  name=“suresh”;
      public  int age=20;
 }
```

what is an object?

Object is a physical entity or a real world entity

Syntax: `Student s = new Student();`

difference between string and StringBuilder?

String	StringBuilder
Strings are immutable	String builder is mutable
new strings are created if we try to modify strings. hence memory wastage will be more.	New strings will not be created when we modify strings. Hence no memory wastage.

what is the purpose of a constructor?

using constructors we can initialize object or we can assign data into object. constructor name must be same as class name.

```
public class Patient
{
    public string name;
    public int age;
    public Patient( string name, int age)
    {
       this.name=name; this.age=age;
    }
}
```

what is a default constructor?

Default constructor is useful for initializing objects with default values.

default constructor will be created when we don’t have constructor in the class.

what is an Object oriented programming language?

object oriented programming language is a language which supports

encapsulation

abstraction

inheritance

polymorphism

what is encapsulation?

Binding related Variables and Methods in a common related place is called as encapsulation.

```
public class Doctor
{
    public string name;
    public int exp;
    public Doctor(string name, int exp)
    {
      this.name=name; this.exp= exp;
    }
    public string SuggestMedicine(string disease)
    {
      //return medicine name based on disease.
    }
}
```

Note: above doctor class having all variables and methods related to doctor. hence we can say that doctor class is following encapsulation.

what is abstraction?

Hiding implementation details and exposing the required details to the users can be called as abstraction.

what is inheritance and why to use it ?

inheritance allows us to use parent class variables and functions in child class. Inheritance reduce code duplication

```
public class A
{
}
public class B : A
{
}
```

Which class is the base class for all classes in C#?

Object class

what is upcasting?

Identifying parent class object reference by using child class object reference

what is downcasting?

Identifying child class object reference using parent class object reference

what is the purpose of new keyword?

it is used for creating object of a class.

new keyword is also used for hiding base class members in derived class.

what is the use of **base** keyword?

using base keyword we can access **base class members** from **derived class.**

```
public class C
{
     public int x=10;
}
```

```
public class D : C
{
     public int x=10;
     public void M1()
     {
        c.wl ( x) ;
        c.wl (base.x);
     }
}
```

what is the output for the following program?

```
public class E
{
      public E()
      {
         C.WL(“Hi”);
      }
}
```

```
public class F : E
{
      public F()
      {
         C.WL(“Hello”);
      }
}
```

```
F  f = new F();
```

what is the purpose of **this** keyword?

this keyword refers to the current instance or current object.

```
public class Patient
{
    public string name;
    public int age;
    public Patient( string name, int age)
    {
        this.name=name;
        this.age=age;
    }
}
```

what is static constructor?

static constructor is used for initializing static variables.

static constructor can’t have access modifier

static constructor can’t have parameters

```
public class Doctor
{
    public static string Hosp_name;
    static Doctor()
    {
        Hosp_name = “apollo hospital “;
    }
}
```

what is the purpose of instance constructor?

instance constructor is used for initializing instance variables.

- what is the difference between static constructor and instance constructor?

static constructor	instance constructor
only one static constructor allowed per class	we can create any no.of instance constructors in a class
Static constructor can’t take parameters	Instance constructor can have parameters.
Static constructor can’t have access modifier	Instance constructor can have access modifier.
Static constructor will be executed only once even if we create multiple objects for a class	Instance constructor will be executed every time when we create object.

what is a namespace?

namespace is used for grouping logically related classes , structs , enums , interfaces etc

```
namespace N1
{
    -- we can write group of related –
    -- classes    structs  interfaces --
}
```

what is namespace aliasing?

assigning a shortcut name for bigger namespace name is called as name space aliasing. name space aliasing will improve code readability.

what is the purpose of access modifiers and explain about each one of them?

access modifiers are used for providing security to the program members . we have 5 access modifiers. that is public , private , protected , internal and protected internal

public members will be accessible from any where.

private members will be accessible only within the declared class.

protected members are accessible only within declared class and also within derived class.

internal members are accessible only within the declared assembly.

protected internal members are accessible within declared assembly and also in the Declared class as well as in derived class.

- what is the default access modifier for namespace level members?
- Internal
- what is the default access modifier for class level members?
- private
- what is the default access modifier for interface level members?
- public

- what is polymorphism?
- When an method is present with the same name in different forms then that method is said to be exhibiting polymorphism
- what are the different types of polymorphism supported in c#?
- Compile time polymorphism/Static/Early bonding

→example for static polymorphism: overloading

1.method overloading

2.constructor overloading

- Runtime polymorphism/Dynamic/Late bonding

→example for dynamic polymorphism : overriding

- what is overloading?
- when 2 or more methods having same name and different signature in the given class then we can say the method is overloaded.

```
public class G
{
   public string M2(int x)
   {
     return “hi”;
   }
   public string M2(int x, int y)
   {
     return “hello”;
   }
}
```

what is a virtual method?

virtual method is a low priority method. we can change the behavior of virtual method in the derived class.

```
public class H
{
    public virtual int M3(int x , int y)
    {
      return  x+y;
    }
}
```

```
public class  I : H
{
    public override int M3(int x , int y)
    {
      return  x*y;
    }
}
```

what is overriding?

overriding is used for changing the behavior of parent class method in the child class.

for overriding a method its name in child & parent class must be same and also the method signature must be same.

```
public class H
{
    public virtual int M3(int x , int y)
    {
      return  x+y;
    }
}
```

```
public class  I : H
{
    public override int M3(int x , int y)
    {
      return  x*y;
    }
}
```

is it possible to override virtual method?

yes

is it possible to override the override method?

yes

is it possible to override the static method?

no

is it possible to overload the static method?

yes

what is a sealed method?

sealed method is a method whose behavior we can’t change in derived class.

we can declare only an override method as sealed method

```
public class H
{
   public virtual int M3(int x , int y)  {  return  x+y;    }
}
public class  I : H
{
   public sealed override int M3(int x , int y)  {   return  x*y;    }
}
```

- difference between static variables and instance variables?

Instance variable	Static variable
Instance variables are object specific	Static variables are class specific
We can access instance variables using object name	We can access static variables using class name

```
public class  J
{
    public static int  x=10;
    public int y=20;
}
J   j1= new J();
Console.WriteLine( j1.y );
Console.WriteLine( J.x );
```

- difference between instance methods and static methods?

Instance method	Static method
Instance method must be called using object name	Static method must be called using class name
Execution of instance method is slower than the static method	Execution of static method is faster then instance method .
Inside instance method we can use this keyword	Inside static method we cant use this keyword.

```
public class  K
{
   public static int M4()  {  return 10;  }
   public int M5()  {  return 20; }
}
K  k1 = new K();
int  r1 = K.M4();
int r2= k1.M5();
```

what is the output for the following program?

```
public class N
{
    static N()
    {
        C.WL(“Hi”);
    }
}
```

```
public class O : N
{
    static O()
    {
        C.WL(“Hello”);
    }
}
```

```
O  o= new O();
```

is it possible to use this keyword for accessing static variables?

no

is it possible to use this keyword for accessing instance variables?

yes

is it possible to access instance variables from static methods?

no

what are the different types of classes supported in c#?

- Instance class
- Static class
- Sealed class
- Nested class
- Abstract class
- Partial class

what is an instance class?

Instance class is a class for which we can create any number of objects

what is a static class?

- static class is a class which can contain only static members
- static class can't participate in inheritance
- we can't create object of static class
- When a class is having only static members it is better to declare class as static class

```
public  static class P
{
      -- it can contain only
      -- static variables
      -- static methods
      -- static constructor
}
```

what is a sealed class?

- Sealed class is a class which will not contain child
- We can create one or more objects for a sealed class
- virtual method are not allowed in sealed class
- abstract methods are not allowed in sealed class.

```
public  sealed class Q
{
}
```

what is a nested class?

nested class is a class which is declared inside another class. we can create object for inner class using outer class name dot inner class name.

```
public  class R
{
    public class S
    {


    }
}
```

what is a partial class?

Using partial class we can split single class Definition into multiple files. During compilation time all partial classes definitions which are present inside same name space will merge into single class.

what is an abstract class?

abstract class is a class which contains **zero or more** abstract methods . we can't create object for an abstract class.

```
public abstract class T
{
  public abstract void m6();
}
```

when to create an abstract class?

It is recommended to create abstract class when we know implementation for some methods and when we don’t know implementation for some methods.

what is an interface?

interface is like class it can contain only un implemented methods. we can’t directly create an object for interface.

```
public interface I1
{
    int m7(int x , int y);
}
public class U : I1
{
   int  I1.m7(int x, int y)
   {
     return x*y;
   }
}
```

```
U  u = new U();
 I1  i1=u;
int  res =     i1.m7(7,8);
```

when to use interface in programming?

when two developers having code dependency we can eliminate the dependency using interfaces. interface will act as a contract between developers.

Is it possible to implement multiple interfaces in a single class?

yes

what are the differences between abstract class and interface?

Abstract class	interface
It supports implemented and un implements methods	It supports only unimplemented methods
Abstract class supports versioning(You can change abstract class without breaking clients code).	Interface will not support versioning(If you change your interface , it will break clients code.)
A class can inherit from only one abstract class.	A class can implement any no.of interfaces.

Is it possible to inherit a class from multiple classes in c#?

no

what is a property?

property is used for reading & modifying data present in private variable.

```
public class  V
{
   private  int  _x = 20 ;
   public  int  X
   {
      get {   return _x; }
      set {   _x = value ; }
   }
}
```

is it possible to change access modifier for both getter or setter in a read write property (explain) ?

- no we can’t change the access modifier for both getter & setter (since at least one must get access modifier from property definition).
- But we can change access modifier either for getter or setter in a read write property.

what are the differences between value type and reference type?

Value type	Reference type
Value type will directly store data in the variable	Reference type will store address of the actual data in the variable
Reading data from value type is faster	Reading data from reference type is slow

what is a structure?

- structure is similar to a class.
- structure is a value type .
- structure can’t contain initialized instance variables.
- structure can’t have parameter less constructor.
- structure can’t have a destructor.
- structure can’t participate in inheritance.

```
public struct W
{
      public int  x;
      public int y;
}
```

```
W  w ;
w.x=10;
w.y=20;
```

what are the differences between class and structure?

Class	structure
class is a reference type	Structure is a value type
Class can contain initialized instance variables	Structure can’t contain initialized instance variables
Class can contain parameter less constructor	Structure can’t contain parameter less constructor
Class can contain a destructor	Structure can’t contain a destructor
Class can participate in inheritance	Structure can’t participate in inheritance

what is the use of ref keyword?

using ref keyword we can pass address of a variable.

what is the use of out keyword?

using out keyword we can pass address of a variable.

what is the difference between ref and out keywords?

Ref	Out
We must initialize ref variables before passing to a function	We must initialize or re-initialize within the called function
Using ref we can't return multiple values	Using out we can return multiple values from a function

what is the difference between for and foreach loop?

foreach loop is faster than for loop. we can't modify data present in a for each loop iteration variable.

what is boxing?

converting a value type to object type is called as boxing.

```
    int   x = 20;
    objcet   o  = x;
```

what is unboxing?

converting object type to value type is called as unboxing.

```
    int   x = 20;
    objcet   o  = x;
    int   y = (int)  o;
```

what is the use of is keyword?

is keyword is used for checking the data type of a variable.

what is the use of as keyword?

as key word is used for data type conversion

what is the difference between const and readonly variable?

Constant	Read only
we must initialize a const variable while declaring.	read only variables can be initialized during declaration or inside a constructor.
Constant variables must be accessed using class name	Read only variables must be accessed using object name

what is an enum?

enum is used for grouping related constants.

enum is a value type.

```
public enum Days
{
   MONDAY, TUESDAY, WEDNESDAY
}
Console.WriteLine( Days.Monday); // op is Monday
Console.WriteLine( (int)Days.Monday); // op is 0
```

what is an exception?

exception is a run time error

what is exception handling?

it is used for avoiding the abrupt termination of program. exception handling is done using try catch blocks

```
try
{
   // code which may give error
}
catch( exception ex  )
{
    // error storing code
}
```

is it possible to write multiple catch blocks for a given try block(explain)?

yes (we can write multiple catch block from child to parent)

is it possible to write try catch block inside catch or finally blocks?

yes

what is the purpose of a finally block?

important code like database connection closing will be usually written in finally block . clr guarantees the execution of finally block code

what is innerexception in c#?

innerexception is a property of exception class. using the innerexception property we can retrieve the previous exception which caused the current exception.

is it possible to write return statement in finally block? no

what is the use of throw key word?

using throw keyword we can re throw the exceptions back to the callers.

msil is platform dependent or independent?

msil is platform independent

JIT Compiler and CLR is platform dependent or independent?

JIT & CLR are platform dependent.

which class is the base class for all exception classes in C#?

exception

what is manifest in an assembly?

manifest is used for storing the current assembly name , version number , public & private key also the culture details.

what is metadata in an assembly?

metadata will store all class names method name property names etc.. present in the assembly. it will also contain the info about which method belongs to which class.

who will call GC?

CLR will call GC when there is a scarcity of the memory.

GC can delete which type of objects?

only managed objects GC can delete

How to delete an un managed object?

in .net we have 2 ways to delete dead un managed objects. 1) using destructor 2) idisposable interface.

what is a destructor ?

destructor is used for destroying un managed resources.

```
public class Y
{
    ~Y()
   {
   }
}
```

what are the difference between destructor and IDisposable Interface Dispose method?

destructor will be called by the GC while deleting the object. Dispose method must be called by programmer after completing the usage of the object.

what is the problem with non generic collections?

type conversions will happen while inserting or reading data from non generic collections.

what is the difference between stack and queue?

stack is used for retrieving data in LIFO order & queue used for reading data in FIFO order.

what are the benefits with Generics?

generics are useful for achieving type safety and re usability.

what is a nullable type?

nullable type is a value type which is useful for storing null values along with respective value type values.

what is an attribute?

attributes are useful for adding custom meta data into the meta data section of the assembly.

what is the use of reflection?

using reflections we can read metadata present in the assembly.

what is a delegate?

delegate is a type safe function pointer.

delegate is used for implementing callback function.

delegates are internally considered as classes.

```
public class Z {   public void M8 (int x, int y) {  }  }
public delegate void D(int x, int y);
D d = new D(Object Address.Method Name)
```

What is the purpose of multicast delegate?

A **delegate** that points to multiple methods is called a **multicast delegate**. The "+" operator adds a function to the **delegate** object and the "-" operator removes an existing function from a **delegate** object.

What is an event?

In c# events are used for implementing notifications. event will store one or more delegate objects. events are usually used in GUI programming.

What is Extension Methods?

extension method is a static method. using extension methods we can add new methods to an existing class without modifying source code.

What is the C# Using block and why should I use it?

using block is used for creating objects

which implements idisposable interface.

using block will be automatically converted

to try finally pattern and dispose method

will be called in the finally block.