How much do you know about Static Class, Constructor and Methods:

----------------------------------------------------------------------------------

**02. Can a static class inherits from other classes or interfaces?** – NO, primarily sealed and abstract

**03. A static class can define an instance constructor or not?** – NO, since instantiation is not allowed.

**04. A static class can be marked as sealed or not?** – NO

**05. Can a static class contains enums?-** YES, but cannot declare enums static

**06. Is it true that a static class cannot be inherited by any class, why?**

- You can also say that creating a static class is the same as creating a normal class with all static members and having a private constructor since a private constructor prevents the class from both being instantiated and inherited.

**07. Can a static class contains a private constructor?** NO, since instantiation is not allowed.

**10. How a static method of a non-static class can be accessed?** – Directly by using class name

**12. Can a static class contains destructor?** - Explicit destructor is not allowed in the static class.

**13. Can you define an indexer in a static class?**

- No indexers are allowed in a static class since an indexer cannot be static because this pointer is not allowed for referencing static members.

**14. A static data member can be defined using the "var" keyword or not?** - NO

**15. Where do static classes and members are stored in the memory?**

- since static members can be accessed directly without creating instances of the class, they must exist in the memory throughout the lifetime of the application; they don’t need to be garbage collected. Therefore, static members are stored in a special memory area called High-Frequency Heap.

**16. If a static class cannot be instantiated then how it is initialized?**

- A static constructor is called automatically. It initializes the class before the first instance is created or any static members are referenced

**17. Can a static method be overridden, why?**

- Static methods can be overloaded but cannot be overridden, since they belong to the class only.

**19. Can you pass any ref or out parameter to a static method or not?** - YES

**20. Can a non-static class contain a static constructor?** - YES

**21. Is parameterized static constructor is allowed?** - NO

**22. Can a static constructor be private?** - NO

**23. How a static constructor is called?**

- A static constructor cannot be called directly and is only meant to be called by the CLR. A static constructor is executed only once in the lifetime of the program or application.

**25. Give me some real examples of static class and method used in .Net.** - System.Math, System.IO.File

**26. What is the use of static class?**

- cases where we need some utility functions and properties. Utility functions are those functions which are very common in some software and everyone in that particular software is making use of those functions. so it does not make sense every time to create an object for such a class which is being used time to time by many programmers.

**28. What is the default access level of a static constructor?** - Private

**29. What is the difference between a static class and a non-static class having all static members with private constructor?** – Can create instance inside the class in case of non-static class

**30. A static field can be defined as const in a static class?** – NO, since a const field is implicitly static in its behavior

**31. A static field can be defined as readonly in a static class?** - YES

**32. Can a structure be defined as static?**

- A structure can also contain static methods,constructor and static data members, but a struct cannot be static itself.

**34. Can a static method declare local static variables?** - NO

**35. When the static constructor is called if you create a delegate to a static method?**

- If a static method is assigned to a delegate or event, then the static constructor is called only when a static method assigned to an event or a delegate is invoked and not when it is assigned.

**36. What will happen if a static constructor throws an exception?**

- the runtime will not invoke it a second time, and the type will remain uninitialized for the lifetime of the application domain.

**37. Can an interface be defined as static in C#?** – NO, but can contain static methods(C# 8)

**38. What is the difference between static class and singleton?**

- Singleton is a design pattern that makes sure that the application creates only a single instance of the class at any time. A static method is used for singleton implementation.

On the other hand for a static class, a single instance remains in memory for the lifetime of the Application Domain in which your program resides and it is accessed globally throughout the application.

**39. The Main() method can be non-static in C #?**- NO

**40.How a sealed class can solve the "fragile class problem" in C #?**

A sealed class can solve the fragile class problem in C# by preventing inheritance.

How much do you know about the Sealed Class in C#

----------------------------------------------------------------------------------------------------------------------------------------

**01. What are the different ways to prevent inheritance in C#?** – SEALED,PRIVATE,STATIC, STRUCT

**06. Can a struct be defined as sealed in C#?** – SEALED by default

**13. Can a sealed class be abstract in C#?** - NO

**18. Can a sealed class implement interface in C#?**

- Yes, a sealed class can implement an interface. But remember a static class is also sealed implicitly, but it cannot implement any interface.

**21. Can a sealed class define sealed methods in C#?**

- Yes, a sealed method can be defined in a sealed class, but the sealed class must be a derived class, and the base class must have virtual or abstract methods for override.

**23. Can a partial class be sealed in C#?** - YES

**24. Can a sealed class be private or a sealed class with all members as private is allowed in C#?**

- No, a sealed class cannot be declared private explicitly. In fact, any class defined in a namespace cannot be explicitly declared as private, protected, protected internal, or private protected.

Yes, a sealed class with all members as private is allowed in C#.

**28. Can you give me some sealed class examples available in the .Net Framework?**

- System.String class ,structs, enums, static classes.

Abstract

----------------------------------------------------------------------------------------------------------------------------------------

**05. What is the different between abstract method and virtual method in C#? Can you use abstract keyword while overriding the abstract method in C#?**

An abstract method cannot contain body definition while a virtual method must have a body definition.

Overriding all the abstract methods in the inheriting concrete class is mandatory while overriding the virtual methods is optional.

The "abstract" modifier can be used while overriding the abstract method but the "virtual" modifier cannot be used while overriding a virtual method.

An abstract method can only be declared in an abstract class while a virtual method can be defined in a non-abstract class as well.

A virtual method can be overridden to abstract for achieving re-abstraction but an abstract method cannot be overridden to virtual.

**07. Can an interface have abstract methods in C#?** - NO

**10. Can you give some real examples of abstract classes and abstract methods available in .Net?**

- System.ValueType, System.Type

Virtual Methods and Method Overriding:

----------------------------------------------------------------------------------------------------------------------------------------------------------

**01. What do you mean by dynamic or late binding in C#?**

– Virtual methods are used to implement dynamic binding. It is also known as late binding or dynamic polymorphism.

**04. Can a static class contains virtual methods?** – NO, abstract, virtual, override not allowed

**07. What is the difference between method overriding and method hiding?**

- Showing is used to protect against subsequent base class modification, while overriding does polymorphism by defining a different implementation.

Shadowing can redefine the entire signature, on the other hand, overriding can redefines only the implementation of a method, not the signature.

**10. If A is a super base class having a virtual method named Fun() which is overridden in the derived class B. Now there is a class C which inherits from B. In that case which implementation of Fun() will be inherited in class C?**

- From the class B, If you have overridden in class C then that will be called.

Constructors

----------------------------------------------------------------------------------------------------------------------------------------------------------

**Q05. If every class has an implicit Default Constructor then why explicit Default Constructor required?**

- That default constructor simply invokes the parameterless constructor of the direct base class. If you have defined only a normal parameterized constructor in a base class and no explicit default constructor, in that case, the derived class cannot call the base class default constructor implicitly. You have to call the parameterized base class constructor explicitly from the derived class.

But in case you have not defined any explicit default constructor and only a parameterized constructor with params type parameters, in that case also the derived class calls the base class default constructor implicitly. There is no need to call the base class parameterized constructor explicitly.

So, the default constructor is always of the form:

public C(): base() {}

protected C(): base() {}

**Q06. If a class has defined an explicit constructor, in that case, the implicit or system defined constructor will be invoked or not?**

**Q09. Can a struct have a Default Constructor in C#?** – No(explicit constructor in Default constructor but only defined)

**Q10. If a base class has 5 different constructors, in that case, is it mandatory to define 5 different constructors in the derived class also?**

**Q02. How can you return a value from a constructor in C#?**

- In the constructor return statement with a value is not allowed but you can have return; statement.

**Q03. Can a constructor have OUT or REF parameters in C#?** - YES

**Q04. Can a constructor have PARAMS or OPTIONAL parameters in C#?** - YES

**Q05. What is Constructor Overloading in C#? Does the order of constructor overloading matter?** - YES

**Q06. Can a constructor call other constructors of the same class?**

**Q07. What is Constructor Chaining in C#? Have you ever used constructor chaining in any project?**

- Constructor Chaining is an approach where a constructor calls another constructor in the same or base class.

**Q08. Is Circular Constructor Chaining allowed in C#?**

**Q09. Is it mandatory for each constructor of the derived class to call the base class constructors?**

- Yes Atleast one must be called

**Q10. Can you throw an exception from a constructor in C#?**

- yes

**What is the difference between implicit and explicit default constructors?**

Explicit means done by the programmer. Implicit means done by the CLR

**Q02. What is the Class Constructor Execution Sequence Order in C#?**

- Static🡪 instance constructor if any

**Q03. What is the Structure Constructor Execution Sequence Order in C#?**

- A structure can contain only a parameterized constructor or a static constructor.

**Q04. What is the Static Constructor Execution Sequence Order in C#?**

- Child to parent

**Q05. What is the Structure Static Constructor Execution Sequence Order in C#?**

- Static🡪 instance constructor if any

**Q07. What is the Static Constructor Execution Sequence Order in the parent-child class hierarchy?**

- Static(child) 🡪 static(parent) 🡪 further normal

**Q08. If the parent class constructor is called first, then what is the order of the class member variables initialization?**

- Static(child) 🡪 initialize variables child 🡪instance ctor(if any) 🡪 static(parent) 🡪 initialize variables parent 🡪instance ctor(if any)

Try Catch Blocks

----------------------------------------------------------------------------------------------------------------------------------------------------------

**Q1. Can you have try block without catch block in C#?** - Yes,but need finally

**Q2. Can you define multiple finally blocks for a single try block in C#?** - NO

**Q3. Can you define multiple catch blocks for a single try block in C#?** – YES

IN,REF,OUT

----------------------------------------------------------------------------------------------------------------------------------------------------------

**Q02. What is the difference between REF, OUT, and IN type parameters in C#?**

* - ref is used to state that the parameter passed *may* be modified by the method.(needs to be initailized)
* in is used to state that the parameter passed *cannot* be modified by the method. (needs to be initailized)
* out is used to state that the parameter passed *must* be modified by the method.

**Q04. Can an IN, REF, or OUT type parameter be optional in C#?** - YES

**Q05. Can two methods be overloaded by the parameter type i.e. IN, REF, and OUT in C#?** - YES

**Q06. Can you pass a Property or Indexer as IN, REF, or OUT type parameter in C#?** - NO

**Q07. Among IN, OUT, and REF which one is bi-directional in C#?** - REF

**Q08. Can IN, OUT or REF type parameters be used in ASYNC methods?**

**Q09. Can IN, OUT or REF type parameters be used in any iterator method which has yield return or yield break?**

**Q10. What is the difference between PassByReference and ReferenceType in C#? When you pass the reference type object to any method, then it is passed as a value or reference in C#?**

Try Catch Blocks

----------------------------------------------------------------------------------------------------------------------------------------------------------

**Q02. What do you mean by the 'Weak Encapsulation' and 'Strong Encapsulation' in OOPS?**

- If class contain any property access level other than private and method access level other than public. Then that class is called as Weak Encapsulated class.

**Q03. Why extends is known as evil in any Object-Oriented Programming Language?**

**Q04. What is the difference between Inheritance and Composition?**

**Q05. Why Composition is immune to Fragile Base Class Problem?**

**Q06. How can you resolve the Fragile Base Class Problems?**