Top C# OOPS Interview Questions and Answers for 2024

1. Define Encapsulation?

Encapsulation is the process of binding/combining the data members and member functions into a single unit.

2. How do you define Abstraction?

It is known as the process of hiding the implementation details and highlighting the essential features.

3. State the meaning of Inheritance.

It is known as the process of deriving a new class from an existing one.

4. What do you understand by Polymorphism?

Polymorphism is known as the situation where a message can be processed in different ways.

5. What are the two types of Polymorphism?

One can define two types of Polymorphism-

* Compile time Polymorphism, also known as Overloading
* Runtime polymorphism, also known as Overriding

8. Define Constructor.

Constructor is known as a unique method of the Class which will be automatically invoked as soon as an instance of the Class is curated. The primary use of the constructor is known to start private fields of the Class as and when creating an instance for the Class.

9. Define the use of Private [Constructor in C#](https://www.simplilearn.com/tutorials/c-sharp-tutorial/c-sharp-constructor).

It is used to pause the object curation of a class. This is used in Singleton Class. It can be used to stop a class from being inherited.

12. Define Virtual?

One uses Virtual to modify a method/property/indexer or event declared in the base class. It allows it to be overridden in the specific derived Class.

13. Define Override in C#?

It extends a virtual method/property/indexer or event of the given base class into a derived class.

14. State the meaning of Interface.

An Interface looks precisely like a class. However, the Interface doesn’t come with any implementation. Things that Interface carries are declarations of events/indexers/properties.

15. Why should one use Interface?

Interfaces are mainly used for Extensibility, Implementation Hiding, Accessing Objects using the interfaces, and Loose Coupling.

16. Define Implicit Interface Implementation.

This is the most common way to implement the members of an interface. In this one, one doesn’t have to specify the interface name of the members and implement the implicity.

17. Define the meaning of Explicit Interface Implementation.

It is known as another way of implementing the members of an interface. In this one, one doesn’t need to specify the interface name of the given members. The difficulty with explicit implementation is that the member of the same class can't be accessed while using a class instance. It can be used only through an instance of the Interface.

18. What do you mean by Encapsulation?

Encapsulation is known as wrapping, also considered as hiding properties and methods. It is utilized for hiding the code and data in one unit to cover the data from the outside world. For Encapsulation, the best example is Class.

19. What do you understand by the term Abstraction?

Abstraction highlights the showcase of only necessary details to the intended user.

20. When can one use Abstract Class?

An Abstract class can be used when one has a requirement where the base class provides the default implementation of the given metaphors. However, other methods must be available to override the child classes. In this situation, one has to use abstract classes.

21. What is the type of members one can define in an Abstract Class?

One can define static and non-static members, including properties, fields, indexers, and abstract classes.

22. Can we restrict object creation in C#?

Yes, one can create an Object Creation by using the following methods-

Abstract Class, Static Class, Private or Protected Constructor.

23. Is it possible to inherit Enum in C#?

No, it is impossible to inherit Enum in C# because the Enums are by default sealed.

24. Define Constant?

Constant is known as the “const” keyword in C#. These are also known as immutable values visible at the compile time. It doesn’t change their values at a run time until and unless the application is running.

25. State the types of Design Patterns.

There are three types of Design Patterns-

Creational Patterns: It deals with the creation of Objects and Classes.

Structural Patterns: It deals with Class and Object Composition.

Behavioral Patterns- It deals with Class and Object Communication.

26. Is it possible to serialize Hashtable?

No, it’s impossible to serialize Hashtable because the .Net Framework restricts the serialization of any objects that implement the iDictionary Interface.

27. How can one catch multiple exceptions at once in C#?

One can use the condition statement in C# to catch multiple exceptions.

28.  Define Accessors.

Accessors are known as the getting and Set portions of a property.

30. Define Base keywords.

One uses base keyword to access members of the base class from a derived class.