1.. What are the six combinations of access modifier keywords and what do they do?

Public: a public member can be accessed from any part of the program.

Protected: a protected member can be accessed within its own class, its subclasses, and any classes within the same package as the declaring class

Private: A private member can only be accessed within the class where it is declared

Default: aka package-private: a member with no explicit access modifier keyword ca only be accessed by other classes within the same package.

Final: a final member cannot be modified once it has been initialized. If applied to a class, the class cannot be subclassed

Static: a static member belongs to the class itself rather than to any specific instance of the class, that means the method declared as static could be invoked without creating an object.

2.What is the difference between the static, const, and readonly keywords when applied to a type member?

Static members belong to the class itself and are initialized once and retain their values until the program terminates, while const and readonly members are used for values that cannot be modified after initialization, but const member are compile-time constants and readonly members can be assigned a value at runtime

3. What does a constructor do?

A constructor is used for initializing a new object. When you create a new Object, the constructor will be invoked and run the code for initializing the object.

4. Why is the partial keyword useful?

Partial key word allows a single class split into multiple class with the same name, in other word, they are treated as same class in compiler.

It is good for organize large class, if one of your class is large and complexity that have a lot of properties and methods. If you declared as partial, it will be well organized and improve code maintainability.

5. What is a tuple?

Tuples are a lightweight data structure that allows you to group multiple elements of different types into a single object. Tuples are similar to arrays or lists but they are immutable and have a fixed size once they are created.

For example Tuple<int, string, double> myTuple = new Typle<int, string, double>(42, “hello”, 3.14);

You can get them one by one.

6. What does the C# record keyword do?

Record allows you to create immutable classes with value semantics in a concise and expressive way. The compiler generates several methods and properties for you automatically, such as constructor, equality. it is easy to create simple data classes that represent data structures and to perform common operations on them.

7. What does overloading and overriding mean?

Overloading refers to the ability to define multiple methods with the same name but different parameters in a class.

Overriding refers to the ability to redefine a method in a subclass that ware already defined in its superclass. The overriding method must have the same name, return type, and the parameters as the original method.

8. What is the difference between a field and a property?

A field is a variable that belongs to a class and stress a value that is associated with an instance of the class. Fields are declared using a field declaration statement and accessed using the dot notation syntax.

The main difference between a field and a property is that fields are accessed directly, while properties are accessed through their accessor method. Properties provide a way to control how the values of fields are accessed and modified, allowing for greater encapsulation and abstraction of the classes’ internal state. Properties can be used to perform validation or other operations when a value is accessed or set.

9. How do you make a method parameter optional?

In C#, you can make a method parameter optional by using optional key word in the parameter declaration. For example:

Public void method(int I, string optionalParam = “”default”){}

When you invoke this method, you can use method(1) or method(1, “hello”);if you don’t give the value for the second parameter, the function will use the value between the double quote as a default parameter.

10. What is an interface and how is it different from abstract class?

Interface is a contract that defines a set of methods, properties, and events that a class must implement. An interface is a completely abstract type that means it does not provide any implementation details. It only specifies the method signatures or properties that must be implemented by any class that implements the interface.

Interface Abstract class

Cannot contain implement code can contains implement code for non-abstract method

Inherit from multiple interface. Inherit from a single abstract class

Must be public can have different access modifiers

Used to define a contract that class must implement. Provide basic implementation

11. What accessibility level are members of an interface?

All member of an interface must be public. That means any class that implements the interface must provide a public implementation of all member in the interface.

12. ✅True/False. Polymorphism allows derived classes to provide different implementations

of the same method.

13. ✅True/False. The override keyword is used to indicate that a method in a derived class is

providing its own implementation of a method.

14. ✅True/False. The new keyword is used to indicate that a method in a derived class is

providing its own implementation of a method.

15.❌True/False. Abstract methods can be used in a normal (non-abstract) class.

16.✅True/False. Normal (non-abstract) methods can be used in an abstract class.

17.✅True/False. Derived classes can override methods that were virtual in the base class.

18.✅True/False. Derived classes can override methods that were abstract in the base class.

19.✅True/False. In a derived class, you can override a method that was neither virtual non abstract in the base class.

20.❌True/False. A class that implements an interface does not have to provide an

implementation for all of the members of the interface.

21.✅ True/False. A class that implements an interface is allowed to have other members that

aren’t defined in the interface.

22.❌True/False. A class can have more than one base class.

23. ✅True/False. A class can implement more than one interface.

24. What is meant by the terms managed resource and unmanaged resource in .NET

“managed resource” refers to any resource that is allocated and managed by the Common Language Runtime. This is includes objects that are created using the .NET framework, such as strings, arrays and collections

“Unmanaged resource” refers to any resource that is not managed by the CLR, include resource such as memory that is allocated using unmanaged APIs or external libraries and handles to system resources such as windows or thread, which must be manually released by the application code to prevent memory leaks and other issues.

25. What's the purpose of Garbage Collector in .NET?

It is to manage memory allocation and deallocation for the applications running on the .NET framework. As the application creates more objects, it continues to allocate more memory, the Garbage Collector will help you to remove unused space in memory and allows these space to store new Objects. The Garbage Collector scan the memory periodically and freeing up the memory which stored some object which is no longer being referenced by your application