
Question: Why can't you create an instance of an interface directly?

- An **interface** is just a contract that defines a set of methods and properties without any implementation.
- Since it doesn't contain actual executable code, you cannot instantiate it with `new IMyInterface()`.
- Instead, you must create a **class** or **struct** that implements the interface, and then you can create an object of that class and reference it via the interface type.

Question: What are the benefits of default implementations in interfaces introduced in C# 8.0?

- They allow you to provide a **default implementation** for interface methods.
- This means when you add a new method to an interface, existing classes don't break because they inherit the default behavior automatically.
- It improves **backward compatibility** and reduces boilerplate code.
- Interfaces can now contain **shared logic**, avoiding repetition across multiple implementations.

Question: Why is it useful to use an interface reference to access implementing class methods?

- It enables **polymorphism**, letting you work with different classes through a single interface type.
- Makes your code more **flexible** and **extensible**—you can swap implementations without changing the code that uses them.
- It's essential for writing **loosely coupled** and **testable** code (e.g., dependency injection).

Question: How does C# overcome the limitation of single inheritance with interfaces?

- C# does not allow multiple class inheritance, but it allows a class to implement **multiple interfaces**.
 - This gives you the benefits of multiple inheritance (reusing contracts and behaviors) without the complexity and ambiguity of diamond inheritance problems.
-

Question: What is the difference between a virtual method and an abstract method in C#?

- **Virtual method:** Has a default implementation in the base class, but derived classes can **override** it if they need different behavior.
 - **Abstract method:** Declared without implementation in an abstract class, and **must** be implemented in any non-abstract derived class.
-