General Interface Interview Questions (C#)

**1. What is an interface in C#?**

An interface is a contract that defines **method/property signatures** without implementation. A class or struct that implements an interface must provide the actual implementation.

interface IShape

{

void Draw();

}

### 2. ****Can interfaces contain fields in C#?****

❌ **No**, interfaces cannot contain fields. They can only contain:

* Method signatures
* Properties
* Events
* Indexers  
  (With default implementations in C# 8+, but not common in interviews)

### 3. ****Can you instantiate an interface directly?****

❌ No. Interfaces cannot be instantiated. They must be implemented by a class.

### 4. ****Why use interfaces in C#?****

* To achieve **abstraction**
* To enable **loose coupling**
* To support **multiple inheritance**
* To make code more testable and maintainable

## 🔁 **Multiple Interface Inheritance**

### 5. ****Does C# support multiple inheritance?****

✅ C# **does not support multiple inheritance with classes**  
✅ C# **does support multiple interface inheritance**

6. **How do you implement multiple interfaces in C#?**

interface IReadable

{

void Read();

}

interface IWritable

{

void Write();

}

class File : IReadable, IWritable

{

public void Read()

{

Console.WriteLine("Reading file...");

}

public void Write()

{

Console.WriteLine("Writing to file...");

}

}

7. **What if two interfaces have the same method signature?**

interface IA { void Show(); }

interface IB { void Show(); }

class Example : IA, IB

{

// Common implementation for both

public void Show()

{

Console.WriteLine("Show implementation");

}

}

### 8. ****What if you want separate implementations for each interface?****

Use **explicit interface implementation**:

class Example : IA, IB

{

void IA.Show()

{

Console.WriteLine("IA Show");

}

void IB.Show()

{

Console.WriteLine("IB Show");

}

}

Execution

IA a = new Example();

a.Show(); // IA Show

IB b = new Example();

b.Show(); // IB Show

## **Multilevel Interface Inheritance**

### 9. ****Can an interface inherit from another interface in C#?****

✅ Yes, interfaces can inherit from one or more interfaces.

interface IBase

{

void BaseMethod();

}

interface IDerived : IBase

{

void DerivedMethod();

}

class Sample : IDerived

{

public void BaseMethod()

{

Console.WriteLine("Base method");

}

public void DerivedMethod()

{

Console.WriteLine("Derived method");

}

}

## **Practical/Behavioral Questions**

### 10. ****When would you use an interface over an abstract class?****

| **Feature** | **Interface** | **Abstract Class** |
| --- | --- | --- |
| Multiple Inheritance | ✅ Yes | ❌ No |
| Fields | ❌ Not allowed | ✅ Allowed |
| Constructors | ❌ Not allowed | ✅ Allowed |
| Use Case | Define a contract (what to do) | Define base behavior (how to do) |
|  |  |  |

11. **How does polymorphism work with interfaces?**

interface IAnimal

{

void Speak();

}

class Dog : IAnimal

{

public void Speak() => Console.WriteLine("Bark");

}

class Cat : IAnimal

{

public void Speak() => Console.WriteLine("Meow");

}

// Polymorphism

IAnimal animal = new Dog();

animal.Speak(); // Bark

animal = new Cat();

animal.Speak(); // Meow