Armen Melkumyan • 1st
Technical / Solutions Architect
4mo •

From .NET C# Technical Interviews: Virtual Functions vs. New Functions

Virtual Functions:

- Used for polymorphism (the core of object-oriented programming).
- Defined with the virtual keyword and overridden in derived classes with override.
- Decides at runtime which method to call based on the actual object type.
- Ideal for scenarios where derived classes need to modify or extend the base class behavior dynamically.

New Functions:

- Used to hide a base class method, not override it.
- Defined with the new keyword in the derived class.
- Decides at compile-time which method to call, based on the reference type.
- Best when the derived method isn't logically connected to the base method.

Key Differences:

virtual enables runtime polymorphism for dynamic behavior.

new hides the base method, sticking to compile-time behavior.

When to Use What?

- Go with virtual when you want flexibility and expect derived classes to extend or modify functionality.
- Use new sparingly, only when you need to introduce entirely separate logic without touching the base class behavior.

#DotNet #CSharp #CodingInterviews

```
// Virtual method example
public class BaseClass
{
    public virtual void Display() => Console.WriteLine("BaseClass");
}

public class DerivedClass : BaseClass
{
    public override void Display() => Console.WriteLine("DerivedClass");
}

BaseClass obj = new DerivedClass();
obj.Display(); // Output: DerivedClass

// New Method example
public class BaseClass
{
    public void Display() => Console.WriteLine("BaseClass");
}

public class DerivedClass : BaseClass
{
    public new void Display() => Console.WriteLine("DerivedClass");
}

BaseClass obj = new DerivedClass();
obj.Display(); // Output: BaseClass
```