**Software Engineer Candidate Assessment  
  
Method Resolution and Overriding in C#**

1.In C#, when a method in a base class is marked as virtual, it can be overridden in a derived class.

2.To correctly override a method in a derived class:

* The method must have the same name, parameters, and return type as the base class method.
* The method must be marked with the override keyword.

**Why the Override Fails in This Case :**

* The base class has this method:

public virtual string speak(int x) { return "silence"; }

* The derived class (Dog) has this method:

public string speak(short x) { return "bow-wow"; }

The issue:

* short and int are different types, so C# treats this as method overloading, not overriding.
* Since override is missing and the parameter type is different, the base class method (speak(int)) is called instead.

Fix:  
To properly override the method, the derived class should match the base class method exactly and use the override keyword:

class Dog : Animal

{

public override string speak(int x) { return "bow-wow"; }

}

Second:

There are couple of issues with the code

1. const cannot be used with reference types like class objects

* In C#, the const modifier can only be used with value types (like int, float, char, etc.) or strings.
* const creates a compile-time constant, which means the value must be known at compile time.
* Since A is a reference type (class), you cannot declare it as const.
* **Fix:** we have to use static read-only instead of const for reference types

2. a is not initialized before being accessed

* Even if we change const to static readonly, the object a must be initialized before assigning values to its properties.
* Currently, a is null, so a.a = 10; will result in a NullReferenceException at runtime.
* **Fix:** Initialize the a object: public static readonly A a = new A();

3. main method signature and format issues

* In C#, the main method should have the correct signature:
* **Fix:** static void Main()