**Sealed And Abstract**

In C#, the sealed keyword is used to prevent inheritance of a class or to prevent overriding of a method. When you mark a class as sealed, it means that it cannot be used as a base class for further inheritance. This implies that other classes cannot inherit from the sealed class. Similarly, when you mark a method as sealed, it prevents derived classes from overriding that method.

in C#, you cannot combine the keywords sealed and abstract together for a class or a method. The sealed keyword is used to prevent inheritance in a class, i.e., to stop other classes from inheriting from it. On the other hand, the abstract keyword is used to define a class or members that must be implemented by derived classes.

**Sealed And Virtual**

In C#, the sealed keyword and the virtual keyword serve different purposes and cannot be used together.

The virtual keyword is used to define a method, property, or indexer in a base class that can be overridden in derived classes. It allows a method to be overridden by providing a new implementation in a derived class.

On the other hand, the sealed keyword is used to prevent further overriding of a method that has been declared as virtual in a base class. When you mark a method with sealed in a derived class, it means that this method cannot be overridden any further in subsequent derived classes.