In C#, static, abstract, and virtual are all **modifiers** that can be used to change the behaviour of a method or property. Here is what they mean:

* static: A static member belongs to the type itself, rather than to any instance of the type. This means that you can call a static method or property without creating an instance of the class. Static members are often used for utility functions or constants that are shared across all instances of a class.
* abstract: An abstract member is a member that does not have an implementation in the class where it is defined. Instead, the implementation is provided by a subclass. Abstract members are often used in abstract classes, which are classes that cannot be instantiated directly but can be used as a base class for other classes.
* virtual: A virtual member is a member that can be overridden by a subclass. When you declare a method or property as virtual, you are saying that subclasses are allowed to provide their own implementation of that member. This is useful when you want to provide a default implementation in a base class, but allow subclasses to customize the behaviour if they need to.