**Modifiers**

**abstract**

The **abstract** modifier indicates that the thing being modified has a missing or incomplete implementation. The abstract modifier can be used with classes, methods, properties, indexers, and events. Use the **abstract** modifier in a class declaration to indicate that a class is intended only to be a base class of other classes. Members marked as abstract, or included in an abstract class, must be implemented by classes that derive from the abstract class.

**virtual**

keyword is used to modify a method, property, indexer, or event declaration and allow for it to be overridden in a derived class. For example, this method can be overridden by any class that inherits it:

**sealed**

 modifier prevents other classes from inheriting from it. the **sealed** modifier on a method or property that overrides a virtual method or property in a base class. This enables you to allow classes to derive from your class and prevent them from overriding specific virtual methods or properties.

**static**

modifier to declare a static member, which belongs to the type itself rather than to a specific object. The **static** modifier can be used with classes, fields, methods, properties, operators, events, and constructors, but it cannot be used with indexers, destructors, or types other than classes. A static class can be used as a convenient container for sets of methods that just operate on input parameters and do not have to get or set any internal instance fields.

**override**

modifier is required to extend or modify the abstract or virtual implementation of an inherited method, property, indexer, or event.

**volatile**

keyword indicates that a field might be modified by multiple threads that are executing at the same time. Fields that are declared **volatile** are not subject to compiler optimizations that assume access by a single thread. This ensures that the most up-to-date value is present in the field at all times.

**async**

Indicates that the modified method, lambda expression, or anonymous method is asynchronous.

**New**

Used to hide an inherited member from a base class member.

**const**

keyword to declare a constant field or a constant local. Constant fields and locals aren't variables and may not be modified.

**readonly**

keyword is a modifier that you can use on fields. When a field declaration includes a **readonly** modifier, assignments to the fields introduced by the declaration can only occur as part of the declaration or in a constructor in the same class.

**Unsafe**

To maintain type safety and security, C# does not support pointer arithmetic, by default. However, by using the [unsafe](https://msdn.microsoft.com/en-us/library/chfa2zb8.aspx) keyword, you can define an unsafe context in which pointers can be used.