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.

Abstract classes have the following features:

* An abstract class cannot be instantiated.
* An abstract class may contain abstract methods and accessors.
* It is not possible to modify an abstract class with the [sealed](https://msdn.microsoft.com/en-us/library/88c54tsw%28v=vs.71%29.aspx) modifier, which means that the class cannot be inherited.
* A non-abstract class derived from an abstract class must include actual implementations of all inherited abstract methods and accessors.

Use the **abstract** modifier in a method or property declaration to indicate that the method or property does not contain implementation.

Abstract methods have the following features:

* An abstract method is implicitly a [virtual](https://msdn.microsoft.com/en-us/library/9fkccyh4%28v=vs.71%29.aspx) method.
* Abstract method declarations are only permitted in abstract classes.
* Because an abstract method declaration provides no actual implementation, there is no method body; the method declaration simply ends with a semicolon and there are no braces ({ }) following the signature. For example:
* public abstract void MyMethod();
* The implementation is provided by an [overriding method](https://msdn.microsoft.com/en-us/library/ebca9ah3%28v=vs.71%29.aspx), which is a member of a non-abstract class.
* It is an error to use the **static** or **virtual** modifiers in an abstract method declaration.

Abstract properties behave like abstract methods, except for the differences in declaration and invocation syntax.

* It is an error to use the **abstract** modifier on a static property.
* An abstract inherited property can be overridden in a derived class by including a property declaration that uses the **override** modifier.

An abstract class must provide implementation for all interface members.

The sealed modifier is used to prevent derivation from a class. A compile-time error occurs if a sealed class is specified as the base class of another class.

A sealed class cannot also be an abstract class.