What a class is

Classes are the user defined data types that represent the **state** and **behaviour** of an object. State represents the properties and **behaviour** is the action that objects can perform.

Classes can be declared using the following access specifiers that limit the accessibility of classes to other classes, however some classes does not require any access modifiers.

1. Public
2. Private
3. Protected
4. Internal
5. Protected internal

To learn the details of access specifiers please refer to the following article of mine:

* [Access Modifiers in C#](https://www.c-sharpcorner.com/blogs/access-modifiers-in-c-sharp1)

For example:

public class Accounts  
{

}

**Some Key points about classes**

* Classes are reference types that hold the object created dynamically in a heap.
* All classes have a base type of **System.Object**.
* The default access modifier of a class is **Internal**.
* The default access modifier of methods and variables is **Private**.
* Directly inside the namespaces declarations of private classes are not allowed.

The following are types of classes in C#:

Timeline

Description automatically generated

What an Abstract class is

An Abstract class is a class that provides a common definition to the subclasses and this is the type of class whose object is not created.

Some key points of Abstract classes are:

* Abstract classes are declared using the abstract keyword.
* We cannot create an object of an abstract class.
* If you want to use it then it must be inherited in a subclass.
* An Abstract class contains both abstract and non-abstract methods.
* The methods inside the abstract class can either have an implementation or no implementation.
* We can inherit two abstract classes; in this case the base class method implementation is optional.
* An Abstract class has only one subclass.
* Methods inside the abstract class cannot be private.
* If there is at least one method abstract in a class then the class must be abstract.

For example:

abstract class Accounts  
{

}

Partial Classes

It is a type of class that allows dividing their properties, methods and events into multiple source files and at compile time these files are combined into a single class.

The following are some key points:

* All the parts of the partial class must be prefixed with the partial keyword.
* If you seal a specific part of a partial class then the entire class is sealed, the same as for an abstract class.
* Inheritance cannot be applied on partial classes.
* The classes that are written in two class files are combined together at run time.

For example:

partial class Accounts  
{

}

Sealed Class

A Sealed class is a class  that cannot be inherited and used to restrict the properties.

The following are some key points:

* A Sealed class is created using the sealed keyword.
* Access modifiers are not applied to a sealed class.
* To access the sealed members we must create an object of the class.

For example:

sealed class Accounts  
{

}

Static Class

It is the type of class that cannot be instantiated, in oher words we cannot create an object of that class using the new keyword, such that class members can be called directly using their class name.

The following are some key points:

* Created using the static keyword.
* Inside a static class only static members are allowed, in other words everything inside the static class must be static.
* We cannot create an object of the static class.
* A Static class cannot be inherited.
* It allows only a static constructor to be declared.
* The methods of the static class can be called using the class name without creating the instance.

For example:

static class Accounts  
{

}