Abstract Class
VS
Interface

Abstract Class

- An abstract class is a special kind of class that cannot be instantiated.An abstract class is only to be sub-classed (inherited from).
- In other words, it only allows other classes to inherit from it but cannot be instantiated. The advantage is that it enforces certain hierarchies for all the subclasses.
- In simple words, it is a kind of contract that forces all the subclasses to carry on the same hierarchies or standards.

Interface

- An interface is not a class.
- It is an entity that is defined by the word Interface.
- An interface has no implementation; it only has the signature or in other words, just the definition of the methods without the body.
- As one of the similarities to Abstract class, it is a contract that is used to define hierarchies for all subclasses or it defines specific set of methods and their arguments. The main difference between them is that a class can implement more than one interface but can only inherit from one abstract class.
- Since C# doesn't support multiple inheritance, interfaces are used to implement multiple inheritance.

Technical differences between Abstract class and Interface

Multiple inheritance :

Interface :	A class may inherit several interfaces.
Abstract class :	A class may inherit only one abstract class.

Default implementation :

Interface :	An interface cannot provide any code, just the signature.
Abstract class :	An abstract class can provide complete, default code and/or just the details that have to be overridden.

Constructor :

Interface :	An interface cannot provide contain constructors.
Abstract class :	An abstract class can contain constructors in it.

Characteristics :

Interface : No characteristics can be defined in interfaces

Abstract class : An abstract class can have characteristics

Access Modifiers :

Interface : An interface cannot have access modifiers for the functions, characteristics etc everything is assumed as public

Abstract class : An abstract class can contain access modifiers for the functions and characteristics

Speed:

Interface : Requires more time to find the actual method in the corresponding classes.

Abstract class : Fast to access

Adding functionality (Versioning) :

Interface : If we add a new method to an Interface then we have to track down all the implementations of the interface and define implementation for the new method.

Abstract class : If we add a new method to an abstract class then we have the option of providing default implementation and therefore all the existing code might work properly.