Answer on Question #50509, Programming, C#

Task:

Define an interface. Briefly describe the benefits of using interfaces

Answer:

An interface contains only the signatures of methods, properties, events or indexers. A class or struct that implements the interface must implement the members of the interface that are specified in the interface definition. In the following example, class ImplementationClass must implement a method named SampleMethod that has no parameters and returns void.

```
interface ISampleInterface
{
   void SampleMethod();
}

class ImplementationClass : ISampleInterface
{
   // Explicit interface member implementation:
   void ISampleInterface.SampleMethod()
   {
      // Method implementation.
   }
   static void Main()
   {
      // Declare an interface instance.
      ISampleInterface obj = new ImplementationClass();
      // Call the member.
      obj.SampleMethod();
   }
}
```

An interface can be a member of a namespace or a class and can contain signatures of the following members:

- Methods
- Properties
- Indexers
- Events

An interface can inherit from one or more base interfaces.

When a base type list contains a base class and interfaces, the base class must come first in the list. A class that implements an interface can explicitly implement members of that interface. An explicitly implemented member cannot be accessed through a class instance, but only through an instance of the interface.

Why Use Interfaces?

- To allow a class to inherit multiple behaviors from multiple interfaces.

- To avoid name ambiguity between the methods of the different classes
- To combine two or more interfaces such that a class need only implement the combined result.
- To allow Name hiding. Name hiding is the ability to hide an inherited member name from any code outside the derived class.

https://www.AssignmentExpert.com