

C# applications for interface

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.

Application 1 :

```
using System;
```

```
interface Marvellous
```

```
{
```

```
    void fun();
```

```
    /*
```

```
    We can not write concrete method in interface
```

```
    public void gun()
```

```
    { }
```

```
    */
```

```
}
```

```
interface Infosystems
```

```
{
```

```
    void gun();
```

```
}
```

```
// Class can implement interface
```

```
class Demo : Marvellous
```

```
{
```

```
    public void fun()
```

```
    {
```

```
        Console.WriteLine("Inside concrete function fun : Demo");
```

```
    }
```

```
}
```

// One class can implement multiple interfaces

class Hello : Marvellous, Infosystems

```
{  
    public void fun()  
    {  
        Console.WriteLine("Inside concrete function fun : Hello");  
    }  
  
    public void gun()  
    {  
        Console.WriteLine("Inside concrete function gun : Hello");  
    }  
}
```

class Program

```
{  
    static void Main(string[] args)  
    {  
        // We can not create object of interface  
        // Marvellous mobj = new Marvellous();  
  
        // We can create reference of interface  
        Marvellous mobj1 = new Demo();  
        mobj1.fun();  
  
        Hello hobj1 = new Hello();  
        hobj1.fun();  
        hobj1.gun();  
    }  
}
```