C# applications for Sealed class & Methods

We can use sealed keyword for class and methods of a class.

When applied to a class, the sealed modifier prevents other classes from inheriting from it.

We can also use the sealed modifier on a method or property that overrides a virtual method or property in a base class.

This enables you to allow classes to derive from your class and prevent them from overriding specific virtual methods or properties.

Application 1 :

```
using System;

sealed class Demo
{
   public void fun()
   {
      Console.WriteLine("Inside Demo fun");
   }
}

/*
We can not inherite sealed class
class Hello : Demo
{

}
*/

// We can apply the concept of aggregation with sealed class
class Marvellous
{
   public int i;

   public Demo dobj;
```

```
    public Marvellous()
    {
        dobj = new Demo();
        i = 11;
    }
}

// Selaled method is the method that we can not override
class Infosystems
{
    public virtual void gun()
    {
        Console.WriteLine("Inside Infosystems gun");
    }

    public virtual void sun()
    {
        Console.WriteLine("Inside Infosystems sun");
    }
}

class MyClass : Infosystems
{
    public override void gun()
    {
        Console.WriteLine("Inside Myclass gun");
    }

    public sealed override void sun()
    {
        Console.WriteLine("Inside Myclass sun");
    }
}

class NewClass : MyClass
{
```

```
    public override void gun()
    {
        Console.WriteLine("Inside Myclass gun");
    }

    /*
    public sealed override void sun()
    {
        Console.WriteLine("Inside Myclass sun");
    }
     */
}

class Program
{
    static void Main(string[] args)
    {
        // Create object of sealed class
        Demo obj1 = new Demo();

        Marvellous obj2 = new Marvellous();
        // Call the method of sealed class
        obj2.dobj.fun();
    }
}
```