Without using Singleton Pattern

-------------------------------------------

using System;

using System.Collections.Generic;

using System.Text;

namespace CreationalDesignPatternDemos

{

public class Program2

{

static void Main()

{

SingletonDemo singletonDemo1 = new SingletonDemo();

singletonDemo1.MyMethod();

SingletonDemo singletonDemo2 = new SingletonDemo();

singletonDemo2.MyMethod();

}

}

}

class SingletonDemo

{

public void MyMethod()

{

Console.WriteLine("This is a method"); ;

}

}

}

1. Add Private Constructor
2. Make class sealed
3. Add a static variable

using System;

using System.Collections.Generic;

using System.Text;

namespace CreationalDesignPatternDemos

{

public class Program3

{

static void Main()

{

//SingletonDemo singletonDemo1 = new SingletonDemo();

//singletonDemo1.MyMethod();

//SingletonDemo singletonDemo2 = new SingletonDemo();

//singletonDemo2.MyMethod();

}

}

sealed class SingletonDemo

{

private SingletonDemo() { }

public static SingletonDemo getinstance = null;

public static SingletonDemo MyObject()

{

if(getinstance == null)

{ getinstance = new SingletonDemo();

return getinstance;

}

else {

return getinstance;

}

}

public void MyMethod()

{

Console.WriteLine("This is a method"); ;

}

}

}

Now, how to call this in Main Method

static void Main()

{

SingletonDemo singletonDemo1 = SingletonDemo.MyObject();

singletonDemo1.MyMethod();

SingletonDemo singletonDemo2 = SingletonDemo.MyObject();

singletonDemo2.MyMethod();

//SingletonDemo singletonDemo1 = new SingletonDemo();

//singletonDemo1.MyMethod();

//SingletonDemo singletonDemo2 = new SingletonDemo();

//singletonDemo2.MyMethod();

}