```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Facade
{
    //By using internal access modifiers, class SubsystemC, SubsystemD and their methods can only be accessed within
    the same file;it cannot be used outside.
    //But, class FacadeSingleton's method OperationC(), OperationD() can be accessed outside of class.
    //Thus, it will hide detail operation of class SubsystemC, SubsystemD but, it can be accessed with OperationC(),
    OperationD().
    //To make access at Facade with only onepoint, Facade made with Singleton.
    internal class SubsystemC
    {
        internal string Operation1()
        {
            return "SubsystemC, Operation1";
        }
    }

    internal class SubsystemD
    {
        internal string Operation1()
        {
            return "SubsystemD, Operation1";
        }

        internal string Operation2()
        {
            return "SubsystemD, Operation2";
        }
    }

    public sealed class FacadeSingleton
    {
        private static readonly FacadeSingleton instance = new FacadeSingleton();
        SubsystemC a = new SubsystemC();
        SubsystemD b = new SubsystemD();
        //instance can be accessed with this method
        public static FacadeSingleton Instance
        {
            get { return instance; }
        }
        private FacadeSingleton() { }

        //wrap around SystemC, SystemD operations
        public void OperationC()
        {
            Console.WriteLine(a.Operation1());
        }

        public void OperationD()
        {
            Console.WriteLine(b.Operation1());
            Console.WriteLine(b.Operation2());
        }
    }
}
```