

# Singleton Design Pattern

By Md. Mahedee Hasan, Software Architect, Leadsoft Bangladesh Ltd.

Source: <http://mahedee.net/singleton-design-pattern/>

## Design Pattern

Design pattern is a solution of known problems. These are strategies of solving commonly occurring problems. A design pattern is not a finish design. It is like a template to solve a problem.

## Singleton Design Pattern

Singleton is a software design pattern. It is restrict to create object more than once. This is actually needed when one object can perform its action in whole system. We frequently use a database connection object to connect with database. We don't need multiple objects to create connection with database and close it. We can use singleton in this scenario.

## Implementation of Singleton by C#

### Step 1: Create EmployeeInfo class

This class is to hold employee information.

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

namespace Singleton
{
    public class EmployeeInfo
    {
        public string EmpName { get; set; }
        public string EmpDesignation { get; set; }
        public int MonthlySalary { get; set; }
    }
}
```

### Step 2: Create a singleton class

Here EmployeeService is a singleton class. Constructor of this class is private so that nobody can create its instance from outside. Instance() is a static method which creates instance of the singleton class. It actually forces that only one instance of the object will be created. Lock() is used to create instance of the singleton pattern in thread safe manner in multi threaded environment. The other two methods are used to add employee information in list and get employee salary. These two are as usual method. Don't mix up Singleton class and Static class. Keep in mind; in static class everything must be static like Method, constructor, properties. But in singleton class it is not required. Hope, you will be clear after the following example.

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace Singleton
{
    /// <summary>
    /// EmployeeService is a singleton class
    /// </summary>
    public class EmployeeService
    {
        //Static object of singleton class
        private static EmployeeService instance;

        private List<EmployeeInfo> lstEmployeeInfo = null;

        /// <summary>
        /// Restrict to create object of Singleton class
        /// </summary>
        private EmployeeService()
        {
            if (lstEmployeeInfo == null)
            {
                lstEmployeeInfo = new List<EmployeeInfo>();
            }
        }

        /// <summary>
        /// The static method to provide global access to the singleton object.
        /// </summary>
        /// <returns>Singleton object of EmployeeService class</returns>
        public static EmployeeService Instance()
        {
            if (instance == null)
            {
                //Thread safe singleton
                lock (typeof(EmployeeService))
                {
                    instance = new EmployeeService();
                }
            }

            return instance;
        }

        /// <summary>
        /// Add employee information to the Employee information list
        /// </summary>
        /// <param name="objEmployeeInfo"></param>
    }
}

```

```

public void AddEmployeeInfo(EmployeeInfo objEmployeeInfo)
{
    lstEmployeeInfo.Add(objEmployeeInfo);
}

/// <summary>
/// Get Salary by Name
/// </summary>
/// <param name="name"></param>
/// <returns>Salary of Employee</returns>
public int GetEmployeeSalaryByName(string name)
{
    int monthlySalary = 0;
    foreach (EmployeeInfo objEmployeeInfo in lstEmployeeInfo)
    {
        if (objEmployeeInfo.EmpName.Contains(name))
            monthlySalary = objEmployeeInfo.MonthlySalary;
    }
    return monthlySalary;
}
}
}

```

### Step 3: Access singleton class

This class creates an instance of singleton class by `EmployeeService objEmployeeService = EmployeeService.Instance();` and access singleton class.

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace Singleton
{
    class Program
    {
        static void Main(string[] args)
        {
            EmployeeInfo objEmpInfo1 = new EmployeeInfo() { EmpName = "Mahedee",
            EmpDesignation = "Senior Software Engineer", MonthlySalary = 00000 };
            EmployeeInfo objEmpInfo2 = new EmployeeInfo() { EmpName = "Kamal",
            EmpDesignation = "Software Engineer", MonthlySalary = 30000 };

            //Create a singleton object
            EmployeeService objEmployeeService = EmployeeService.Instance();

            objEmployeeService.AddEmployeeInfo(objEmpInfo1);
            objEmployeeService.AddEmployeeInfo(objEmpInfo2);
        }
    }
}

```

```
Console.WriteLine(objEmpInfo2.EmpName + " : " +  
objEmployeeService.GetEmployeeSalaryByName("Kamal"));
```

```
Console.ReadLine();
```

```
    }  
}
```

MAHEDDEE.NET