**Difference between Static and Singleton**

|  |  |
| --- | --- |
| **Static** | **Singleton** |
| We don’t need to instantiate a static class and can access properties and methods with class name. | Instance creation is necessary to access methods and properties. We create instance using static property defined for this purpose. |
| Static class cannot be passed a parameter in another class | We can pass instances of a singleton as a parameter to another class method. |
| Static keyword is required in all the methods defined in Static class. | The Singleton class does not require you to use the static keyword everywhere. |
| Static class does not support interface inheritance. It throw error “static classes cannot implement interfaces” | Singleton class can implement interface |
| Static variable does not reset until server is restarted. |  |
| It cannot have instance members. We need to call all the static methods explicitly. | This may have instance members. |
| A Static Class cannot be extended | A singleton class can be extended. |

**Reference:**

From <[*http://dotnet-concept.com/Articles/2019/12/5799870/Difference-between-Static-Class-and-Singleton-Pattern-Csharp*](http://dotnet-concept.com/Articles/2019/12/5799870/Difference-between-Static-Class-and-Singleton-Pattern-Csharp)>

From <[*https://www.c-sharpcorner.com/UploadFile/8911c4/singleton-design-pattern-in-C-Sharp/*](https://www.c-sharpcorner.com/UploadFile/8911c4/singleton-design-pattern-in-C-Sharp/)>

From <[*https://stackoverflow.com/questions/519520/difference-between-static-class-and-singleton-pattern*](https://stackoverflow.com/questions/519520/difference-between-static-class-and-singleton-pattern)>