**Singleton Pattern**

In software engineering, the singleton pattern is a software design pattern that restricts the instantiation of a class to one object.

This is useful when exactly one object is needed to coordinate actions across the system. The concept is sometimes generalized to systems that operate more efficiently when only one object exists, or that restrict the instantiation to a certain number of objects. The term comes from the mathematical concept of a singleton.

Ensure a class has only one instance and provide a global point of access to it.

* **Example 1 - Logger Classes**
  + The Singleton pattern is used in the design of logger classes. This classes are usually implemented as a singletons and provides a global logging access point in all the application components without being necessary to create an object each time a logging operations is performed.
* **Example 2 - Configuration Classes**
  + The Singleton pattern is used to design the classes which provides the configuration settings for an application. By implementing configuration classes as Singleton not only that we provide a global access point, but we also keep the instance we use as a cache object.
* **Example 3 - Accessing resources in shared mode**
* **Example 4 – Factories implemented as Singletons**
  + Let's assume that we design an application with a factory to generate new objects.

