**Singleton: -**

The singleton pattern is used to limit creation of a class to only one object. This is beneficial when one (and only one) object is needed to coordinate actions across the system. It is a Creational Design Pattern.

In terms of practical use Singleton patterns are used in logging, configuration settings.

* Exception/Information logging
* Connection pool management
* File management
* Device management such as printer spooling
* Application Configuration management
* Cache management
* And Session based shopping cart

Implementation guidelines: -

## Constructor is Private so that it can't be instantiated from outside of the class.

## We will have the Class's static reference variable and by default it will be NULL.

## We will provide a public static method which will create & return an instance of the class. We will check whether the reference variable is NULL or not before creating an instance of the same.

**Important point:** The Singleton design pattern is intrinsically wrong and should never be used. It mixes in itself the creational logic (it creates itself) and the business logic, not to mention that it holds itself as static global object that it will never make it to the garbage collector (a singleton, like love, is forever) and constitutes a nasty global state.

e.g.

