# Singleton patton (Most used)

The **Singleton Design Pattern** is a creational pattern that ensures a class has only one instance and provides a global point of access to that instance. This pattern is useful when exactly one object is needed to coordinate actions across the system.

## Key Concepts:

- **Single Instance**: The class restricts the instantiation of itself to only one instance.
- **Global Access Point**: Provides a way to access this instance from any part of the code.
- **Lazy Initialization (optional)**: The instance is created only when it is needed for the first time.

## Use Cases:

- **Configuration Management**: Centralized access to configuration settings.
- **Logging**: Ensuring all parts of an application write to the same log file.
- **Resource Management**: Managing access to resources such as databases or hardware devices.

(1) Common logger for entire application

(2) Singleton should be thread safe as well.