**Mandatory Hands-ON:**

**Exercise 1: Implementing the Singleton Pattern**

The **Singleton Pattern** ensures that a **class has only one instance**, and provides a **global point of access** to that instance.

We are simulating a **Logger utility class** and using Singleton pattern to ensure:

**Only one Logger object** exists during the whole application execution

All implementations of the Singleton have these two steps in common:

* Make the default constructor private, to prevent other objects from using the new operator with the Singleton class.
* Create a static creation method that acts as a constructor. Under the hood, this method calls the private constructor to create an object and saves it in a static field. All following calls to this method return the cached object.

Refer repository for code.

OUTPUT:

