DESIGN PATTERNS AND PRINCIPLE : HANDS-ON (MANDATORY)

EXERCISE –1 : SINGLETON PATTERN

- It is to design a system that creates exactly one instance of the logger class and provides a global access point to it .

- Multiple objects could result in duplicate or inconsistent logging.

✓ **CONCEPT :**

1. **SINGLETON :**

* A design pattern that ensures a class has only one instance and provides a global access point to it .

* Used in : Logging , Configuration setting , Database connection management.

✓ **KEYWORDS :**

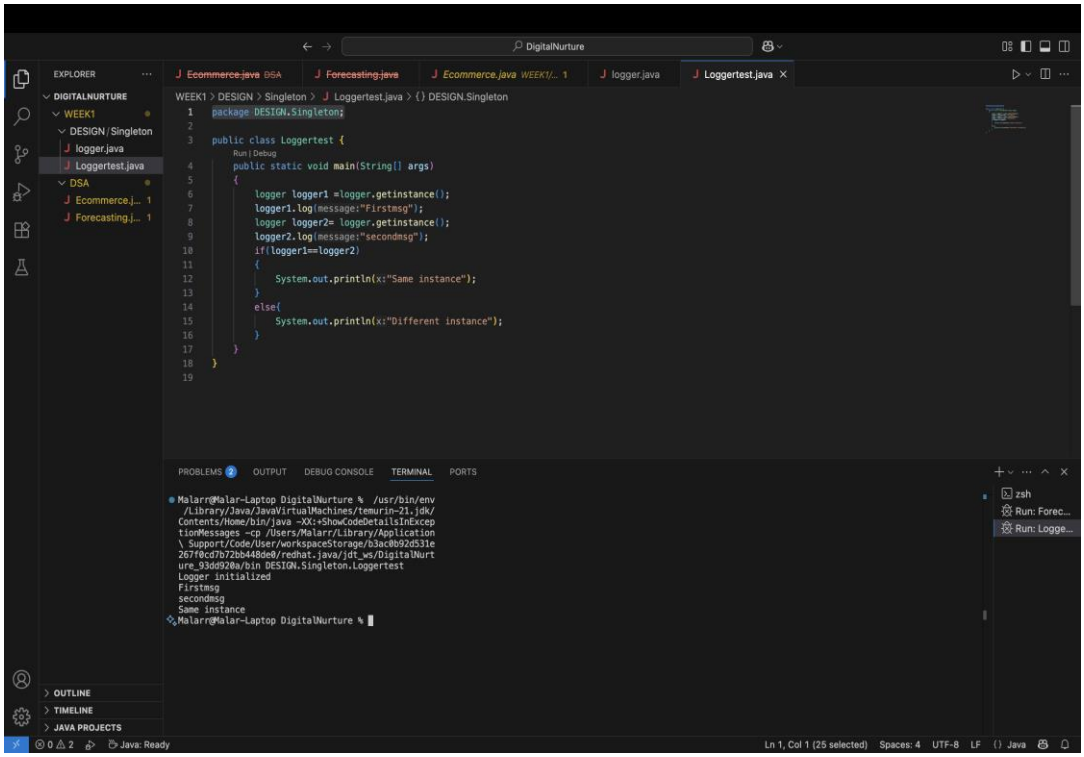
1. *Private* : Restricts access and external object creation.

2. *Instance* : One and only object of the class.

3. *getInstance()* : Method tat controls and provides access to the singleton object.

✓ **TIME COMPLEXITY** : O(1)

✓ **OUTPUT :**



EXERCISE – 2 : FACTORY METHOD PATTERN

- To create a system that handles multiple type of documents.
- Single creation point that knows how to create the correct document depending on the type.

KEYWORDS :

1. *Interface* : Defines what a class must do (without saying how)

2. *Abstract class* : A base class that cant be instantiated but is extended by other classes.

3. *createDocument()* : Factory method that creates and returns the correct document object.

4. *Implements / extends :* Used to implement interface or extend classes for specific behaviour

- ✓ **<u>TIME COMPLEXITY :</u>** *createDocument()* -> O(1)

- ✓ **<u>OUTPUT :</u>**