**Superset ID:6413402**

**SINGLETON PATTERN IMPLEMENTATION**

**Design Patterns and Principles**

Exercise 1: Implementing the Singleton Pattern

**Code implementation**

public class Singleton {

private static Singleton onlyInstance;

private Singleton() {

System.out.println("Singleton instance created!");

}

public static Singleton getInstance() {

if (onlyInstance == null) {

System.out.println("Creating Singleton for the first time...");

onlyInstance = new Singleton();

} else {

System.out.println("Reusing existing Singleton instance...");

}

return onlyInstance;

}

public void showMessage(String user) {

System.out.println("Hello " + user + "! Singleton says hi.");

}

public static void main(String[] args) {

Singleton firstUser = Singleton.getInstance();

firstUser.showMessage("Alice");

Singleton secondUser = Singleton.getInstance();

secondUser.showMessage("Bob");

if (firstUser == secondUser) {

System.out.println("It's the same Singleton instance.");

} else {

System.out.println("Different instances created!");

}

}

}

**OUTPUT**

Creating Singleton for the first time...

Singleton instance created!

Hello Alice! Singleton says hi.

Reusing existing Singleton instance...

Hello Bob! Singleton says hi.

It's the same Singleton instance.

