**Singleton**

**What it does: Ensures that only one class instance is created.**

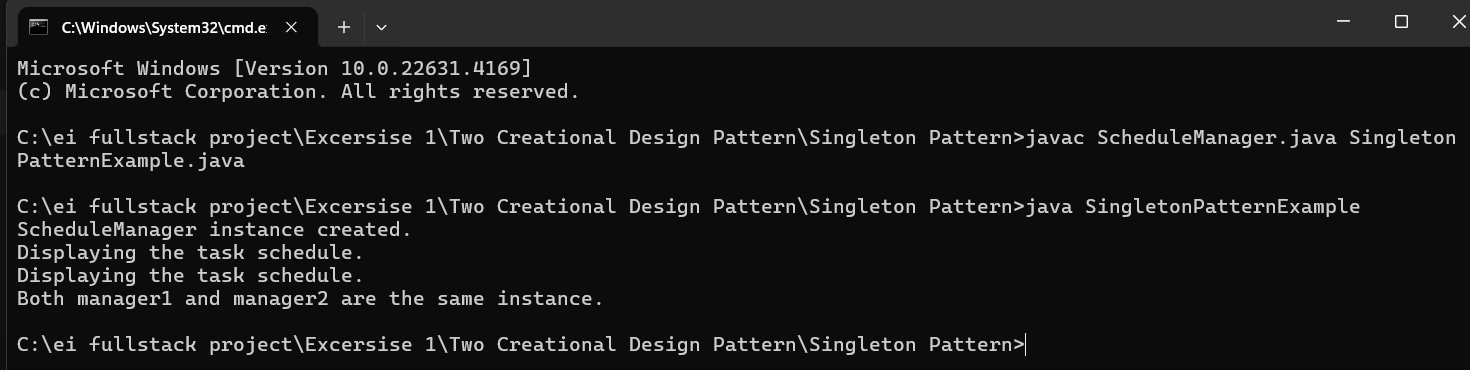
### **1. Problem Statement:**

Design a task management system that ensures there is only one instance of the ScheduleManager class to manage tasks. This Singleton pattern will ensure that all parts of the program access the same instance, preventing the creation of multiple instances. The system should provide a method to display the task schedule and demonstrate that only one instance of ScheduleManager is used.

### **2. Explanation of the Code:**

1. Singleton Pattern:
   * The ScheduleManager class implements the Singleton pattern, ensuring only one instance of the class exists throughout the program.
   * Private constructor: Prevents external instantiation.
   * getInstance() method: Provides the only way to access the single instance. If no instance exists, it creates one. If an instance already exists, it returns that.
2. Main Class (SingletonPatternExample):
   * Demonstrates how to retrieve the instance of ScheduleManager using getInstance().
   * Shows that both manager1 and manager2 refer to the same instance, verifying the Singleton pattern.

**3. Output**

****