**Exercise 1: Implementing the Singleton Pattern**

1. **Basic Singleton Implementation**:
   * The Logger class has a private constructor to prevent direct instantiation
   * It maintains a static reference to its single instance
   * The getInstance() method provides controlled access to the single instance
2. **Testing**:
   * The test class verifies that multiple calls to getInstance() return the same object
   * It includes a simple multi-threaded test
3. **Thread-Safe Variant**:
   * The ThreadSafeLogger uses double-checked locking with a volatile variable
   * This ensures thread safety while minimizing synchronization overhead
4. **Enum Singleton**:
   * The EnumLogger shows the recommended way to implement singletons in Java
   * Enum singletons are inherently thread-safe and serialization-safe

