**Exercise 1: Implementing the Singleton Pattern**

**Scenario:**

You need to ensure that a logging utility class in your application has only one instance throughout the application lifecycle to ensure consistent logging.

**Code:**

**package** SingletonPatternExample;

**public** **class** Logger {

**private** **static** Logger *instance*;

**private** Logger() {

System.***out***.println("Private constructor");

}

**public** **static** Logger getInstance() {

**if** (*instance* == **null**) {

*instance* = **new** Logger();

}

**return** *instance*;

}

**public** **void** print(String message) {

System.***out***.println("Message : " + message);

}

}

**package** SingletonPatternExample;

**public** **class** Main {

**public** **static** **void** main(String[] args) {

Logger logger1 = Logger.*getInstance*();

logger1.print("Hello!How are you?");

Logger logger2 = Logger.*getInstance*();

logger2.print("Have a nice day");

**if** (logger1 == logger2) {

System.***out***.println("Both logger instances are the same.");

} **else** {

System.***out***.println("Logger instances are different!");

}

}

}

**Output:**

