**Exercise 1: Implementing the Singleton Pattern**

**package** com.javacognizant;

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

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

// here we are constructing Private constructor to prevent instantiation

**private** Logger() {

System.***out***.println(" here the Logger Initialized.");

}

// Public static method to provide global access point

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

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

*instance* = **new** Logger(); // Lazy initialization

}

**return** *instance*;

}

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

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

}

// Main method for testing the singleton behavior

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

// Get the singleton instance of Logger

Logger L1 = Logger.*getInstance*();

L1.log("This is the first log message.");

// Get the same instance again

Logger L2 = Logger.*getInstance*();

L2.log("This is the second log message.");

// Checking the if both instances are the same or not

**if** (L1 == L2) {

System.***out***.println("Both logger1 and logger2 refer to the same instance.");

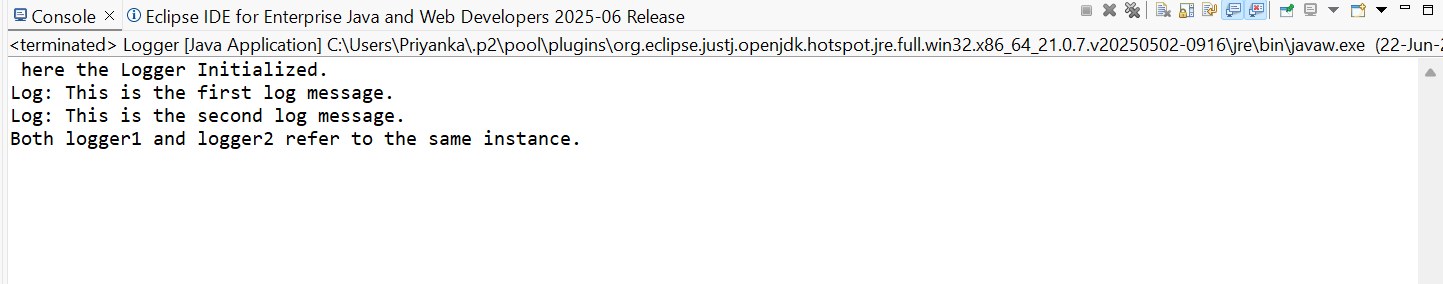
} **else** {

System.***out***.println("Different instances exist,Singleton failed.");

}

}

}

**OUTPUT:**