**Algorithms and Data Structures**

**Exercise 2: E-commerce Platform Search Function**

**Objective:**  
To create different types of documents (Word, PDF, Excel) using the Factory Method Pattern for flexible and scalable object creation.

**Step-by-Step Implementation:**

**1. Project Setup:**

* File Name: SingletonLoggerDemo.java

**2. Code:**

public class SingletonLoggerDemo {

// Static nested Singleton Logger class

static class Logger {

private static Logger instance;

// Private constructor to prevent instantiation

private Logger() {

System.out.println("Logger Initialized");

}

// Public method to provide access to the instance

public static Logger getInstance() {

if (instance == null) {

instance = new Logger();

}

return instance;

}

// Logging method

public void log(String message) {

System.out.println("[LOG]: " + message);

}

}

// Main method to test singleton behavior

public static void main(String[] args) {

Logger logger1 = Logger.getInstance(); // Should initialize the logger

Logger logger2 = Logger.getInstance(); // Should return existing instance

logger1.log("This is the first log message.");

logger2.log("This is the second log message.");

if (logger1 == logger2) {

System.out.println("Both logger instances are the same (singleton works!).");

} else {

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

}

}

}

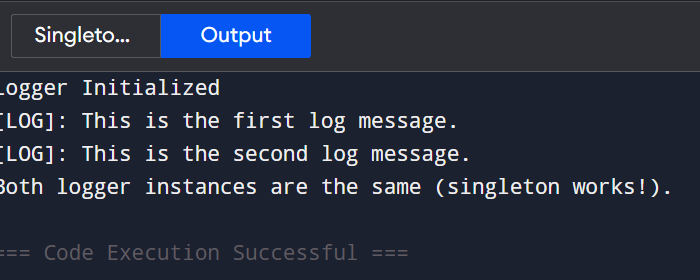
**Expected Output:**

Logger Initialized

[LOG]: This is the first log message.

[LOG]: This is the second log message.

Both logger instances are the same (singleton works!).

**Screenshots:**  


**Exercise 7: Financial Forecasting**

**Objective:** To predict future financial values using recursion based on past growth rates.

**Step-by-Step Implementation:**

**1. Project Setup:**

* File Name: FinancialForecast.java

**2. Code:**

public class FinancialForecast {

// Recursive method to calculate future value

public static double forecast(double initialValue, double growthRate, int years) {

if (years == 0) {

return initialValue;

}

return forecast(initialValue, growthRate, years - 1) \* (1 + growthRate);

}

public static void main(String[] args) {

double initialValue = 10000; // Starting value

double growthRate = 0.05; // 5% annual growth

int years = 5; // Forecast for 5 years

double futureValue = forecast(initialValue, growthRate, years);

System.out.printf("Predicted value after %d years: ₹%.2f%n", years, futureValue);

}

}

**Expected Output:**

Predicted value after 5 years: ₹12762.82

**Screenshots:**

