**WEEK-1 DESIGN PATTERNS AND PRINCIPLES**

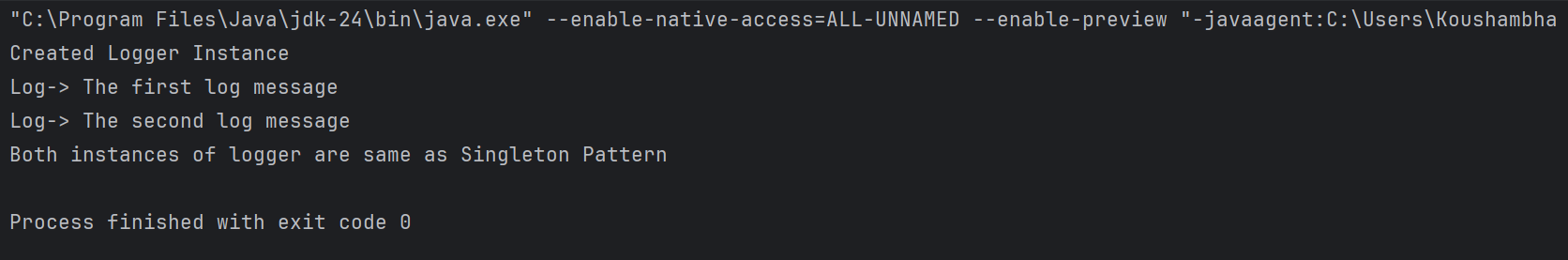
**Exercise 1: Implementing the Singleton Pattern**

**CODE:**

**public class Logger {  
  
  
 private static Logger *instance* = new Logger();  
  
  
 private Logger() {  
 System.*out*.println("Created Logger Instance");  
 }  
  
  
 public static Logger getInstance() {  
 return *instance*;  
 }  
  
  
 public void log(String message) {  
 System.*out*.println("Log-> " + message);  
 }  
}**

**public class Main {  
 public static void main(String[] args) {  
  
 Logger logger1 = Logger.*getInstance*();  
 logger1.log("The first log message");  
  
 Logger logger2 = Logger.*getInstance*();  
 logger2.log("The second log message");  
  
 // Check if both references point to the same object  
 if (logger1 == logger2) {  
 System.*out*.println("Both instances of logger are same as Singleton Pattern");  
 } else {  
 System.*out*.println("Singleton failed");  
 }  
 }  
}**

OUTPUT:



**Exercise 2: Implementing the Factory Method Pattern**

**CODE:**

**package document;  
  
public interface Document {  
 void open();  
}**

**package document;  
public class ExcelDocument implements Document {  
 @Override  
 public void open() {  
 System.*out*.println("Excel Document is opened");  
 }  
}**

**package document;  
  
public class PdfDocument implements Document {  
 @Override  
 public void open() {  
 System.*out*.println("Pdf Document is opened");  
 }  
}**

**package document;  
  
public class WordDocument implements Document {  
 @Override  
 public void open() {  
 System.*out*.println("Word Document is opened");  
 }  
}**

**package factory;  
  
import document.Document;  
  
public abstract class DocumentFactory {  
 public abstract Document createDocument();  
}**

**package factory;  
  
import document.Document;  
import document.ExcelDocument;  
  
public class ExcelDocumentFactory extends DocumentFactory {  
 @Override  
 public Document createDocument() {  
 return new ExcelDocument();  
 }  
}**

**package factory;  
  
import document.Document;  
import document.PdfDocument;  
  
public class PdfDocumentFactory extends DocumentFactory {  
 @Override  
 public Document createDocument() {  
 return new PdfDocument();  
 }  
}**

**package factory;  
  
import document.Document;  
import document.WordDocument;  
  
public class WordDocumentFactory extends DocumentFactory {  
 @Override  
 public Document createDocument() {  
 return new WordDocument();  
 }  
}**

**import document.Document;  
import factory.\*;  
  
public class Main {  
 public static void main(String[] args) {  
 DocumentFactory wordFactory = new WordDocumentFactory();  
 Document wordDoc = wordFactory.createDocument();  
 wordDoc.open();  
  
 DocumentFactory pdfFactory = new PdfDocumentFactory();  
 Document pdfDoc = pdfFactory.createDocument();  
 pdfDoc.open();  
  
 DocumentFactory excelFactory = new ExcelDocumentFactory();  
 Document excelDoc = excelFactory.createDocument();  
 excelDoc.open();  
 }  
}**

OUTPUT:

