***WEEK 1***

***MANDATORY HANDS ON***

***Design Patterns and Principles***

## Logger.java

public class Logger {

private static Logger instance;

private Logger() {

System.out.println("Logger instance created.");

}

public static Logger getInstance() {

if (instance == null) {

instance = new Logger();

}

return instance;

}

public void log(String message) {

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

}

}

### TestSingleton.java

public class TestSingleton {

public static void main(String[] args) {

Logger logger1 = Logger.getInstance();

logger1.log("Starting...");

Logger logger2 = Logger.getInstance();

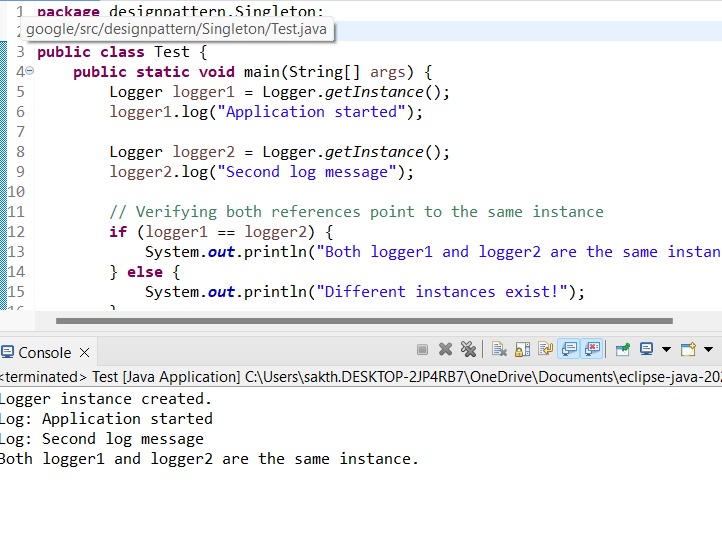
logger2.log("Running...");

System.out.println(logger1 == logger2 ? "Same instance." : "Different instances.");

}

}

### Output



## ****Exercise 2: Factory Method Pattern****

### Document.java

public interface Document {

void open();

}

**WordDocument.java**

public class WordDocument implements Document {

public void open() {

System.out.println("Opening Word document.");

}

}

### PdfDocument.java

public class PdfDocument implements Document {

public void open() {

System.out.println("Opening PDF document.");

}

}

### ExcelDocument.java

public class ExcelDocument implements Document {

public void open() {

System.out.println("Opening Excel document.");

}

}

### DocumentFactory.java

public abstract class DocumentFactory {

public abstract Document createDocument();

}

### WordFactory.java / PdfFactory.java / ExcelFactory.java

public class WordFactory extends DocumentFactory {

public Document createDocument() {

return new WordDocument();

}

}

public class PdfFactory extends DocumentFactory {

public Document createDocument() {

return new PdfDocument();

}

}

public class ExcelFactory extends DocumentFactory {

public Document createDocument() {

return new ExcelDocument();

}

}

### TestFactory.java

public class TestFactory {

public static void main(String[] args) {

DocumentFactory factory = new PdfFactory();

Document doc = factory.createDocument();

doc.open();

}

}

**Output**

