**DESIGN PATTERNS AND PRINCIPLES**

**Exercise 1: Implementation of the Singleton Pattern**

**Code:**

class Logger {

private static Logger instance = new Logger();

private Logger() {

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

}

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("Starting the application.");

Logger logger2 = Logger.getInstance();

logger2.log("Application is running.");

// Check if both references point to the same object

if (logger1 == logger2) {

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

} else {

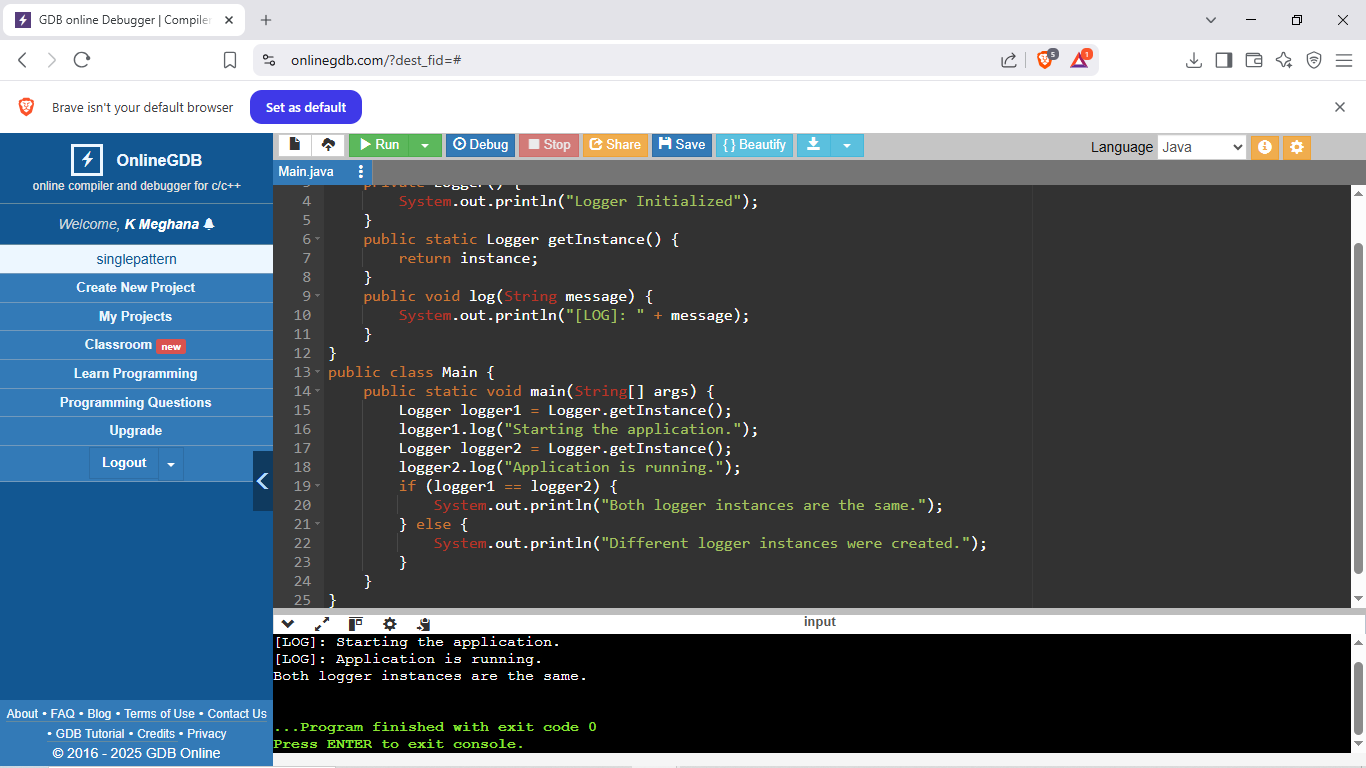
System.out.println("Different logger instances were created.");

}

}

}

**Output:**



In Singleton Pattern, Object is created only once and all variables share same object.

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

**Code:**

interface Document {

void open();

}

class WordDocument implements Document {

public void open() {

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

}

}

class PdfDocument implements Document {

public void open() {

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

}

}

class ExcelDocument implements Document {

public void open() {

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

}

}

abstract class DocumentFactory {

public abstract Document createDocument();

}

class WordDocumentFactory extends DocumentFactory {

public Document createDocument() {

return new WordDocument();

}

}

class PdfDocumentFactory extends DocumentFactory {

public Document createDocument() {

return new PdfDocument();

}

}

class ExcelDocumentFactory extends DocumentFactory {

public Document createDocument() {

return new ExcelDocument();

}

}

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();

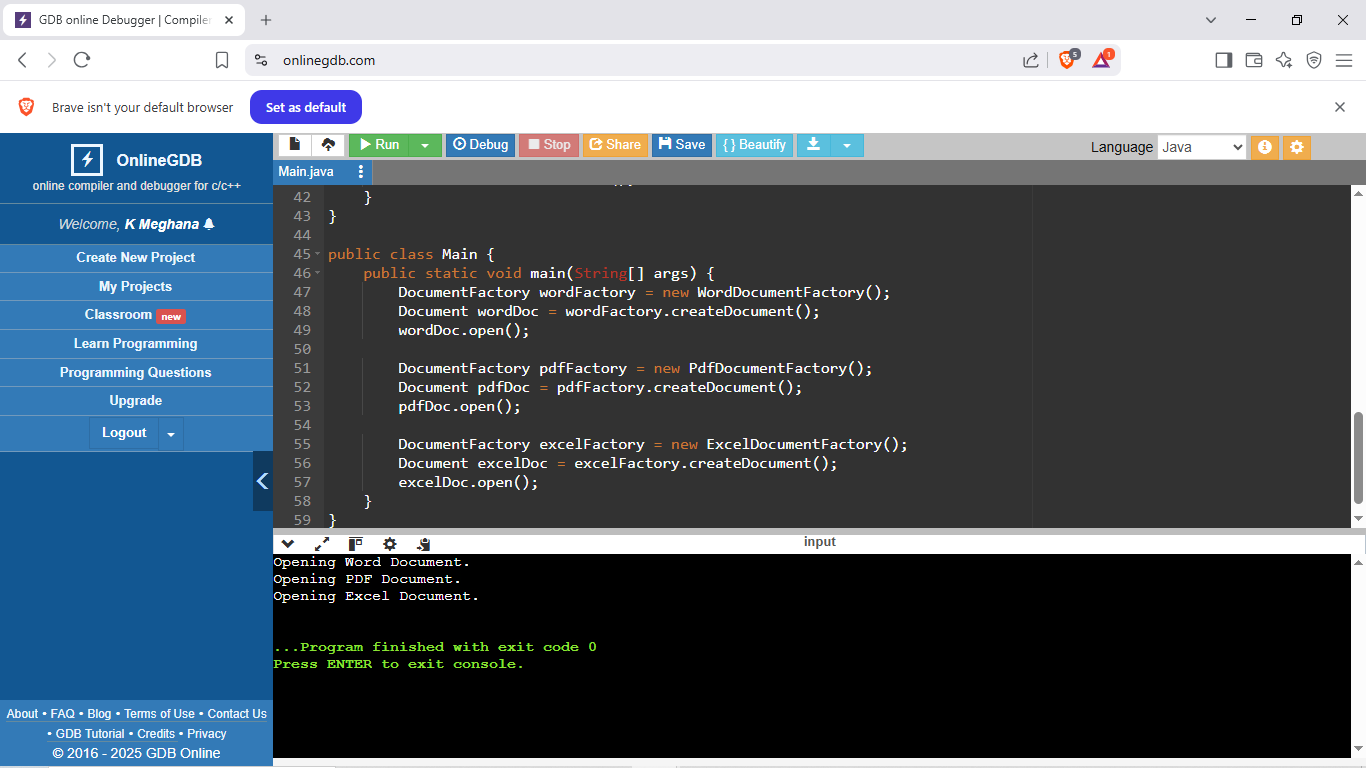
Document excelDoc = excelFactory.createDocument();

excelDoc.open();

}

}

**Output:**



The Factory Method Pattern creates different objects using different factories.