**Exercise 1: Implementing the Singleton Pattern**

**Code:**

public class Main {

static class Logger {

private static Logger instance;

private Logger() {

System.out.println("Constructor called");

}

public static Logger getInstance() {

if (instance == null) {

instance = new Logger();

}

return instance;

}

public void log(String message) {

System.out.println( message);

}

}

public static void main(String[] args) {

Logger logger1 = Logger.getInstance();

logger1.log("Logging from object 1");

Logger logger2 = Logger.getInstance();

logger2.log("Logging from object 2");

if (logger1 == logger2) {

System.out.println("Singleton confirmed: logger1 and logger2 are same");

} else {

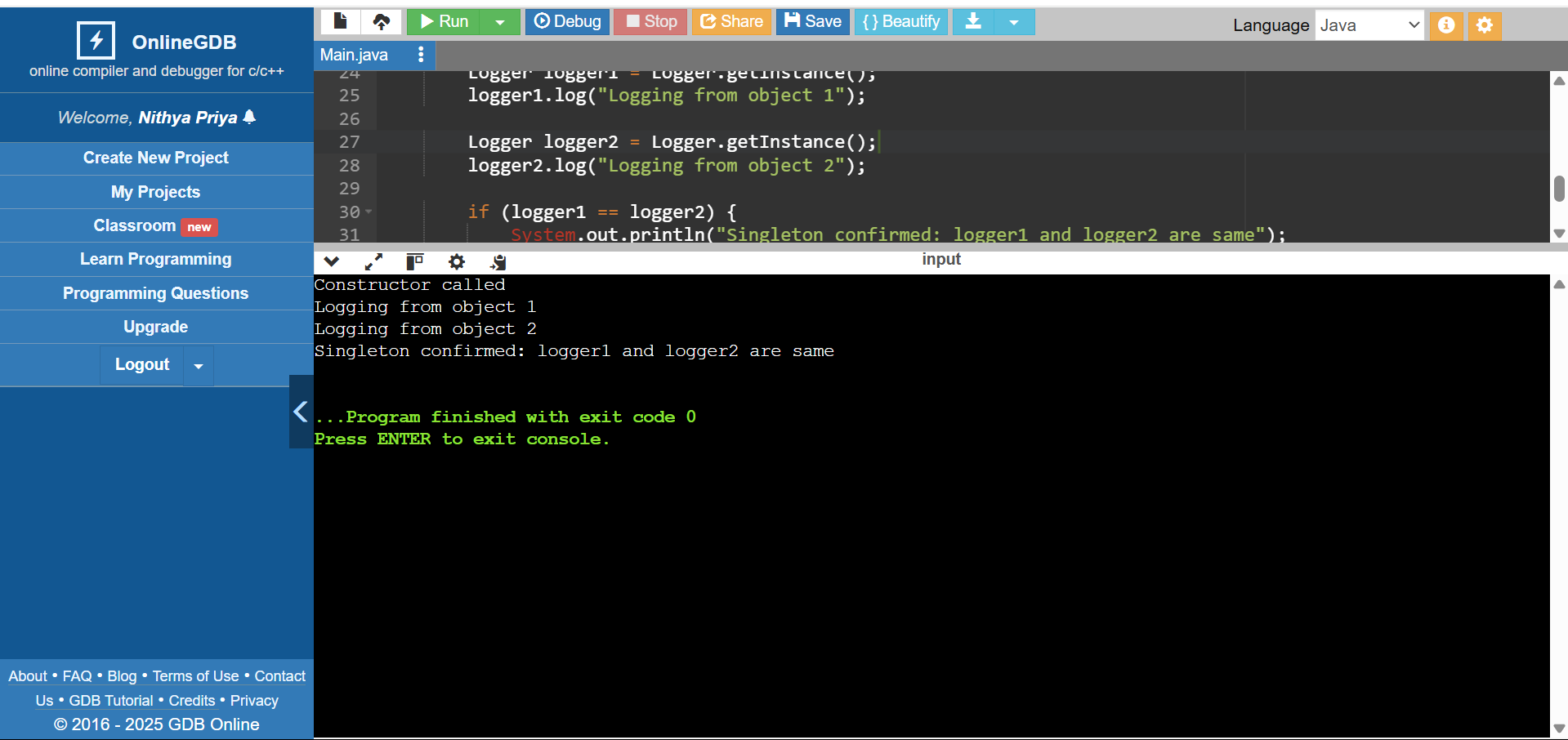
System.out.println("Singleton failed: Different objects");

}

}

}

**Output:**

****

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

**Code:**

public class Main {

interface Document {

void open();

}

static class WordDocument implements Document {

public void open() {

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

}

}

static class PdfDocument implements Document {

public void open() {

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

}

}

static class ExcelDocument implements Document {

public void open() {

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

}

}

static class DocumentFactory {

public Document createDocument(String type) {

if (type.equalsIgnoreCase("word")) {

return new WordDocument();

} else if (type.equalsIgnoreCase("pdf")) {

return new PdfDocument();

} else if (type.equalsIgnoreCase("excel")) {

return new ExcelDocument();

}

return null;

}

}

public static void main(String[] args) {

DocumentFactory factory = new DocumentFactory();

Document doc1 = factory.createDocument("word");

doc1.open();

Document doc2 = factory.createDocument("pdf");

doc2.open();

Document doc3 = factory.createDocument("excel");

doc3.open();

}

}

**Output:**

