# Design principles & Patterns:

Exercise 1: Implementing the Singleton Pattern

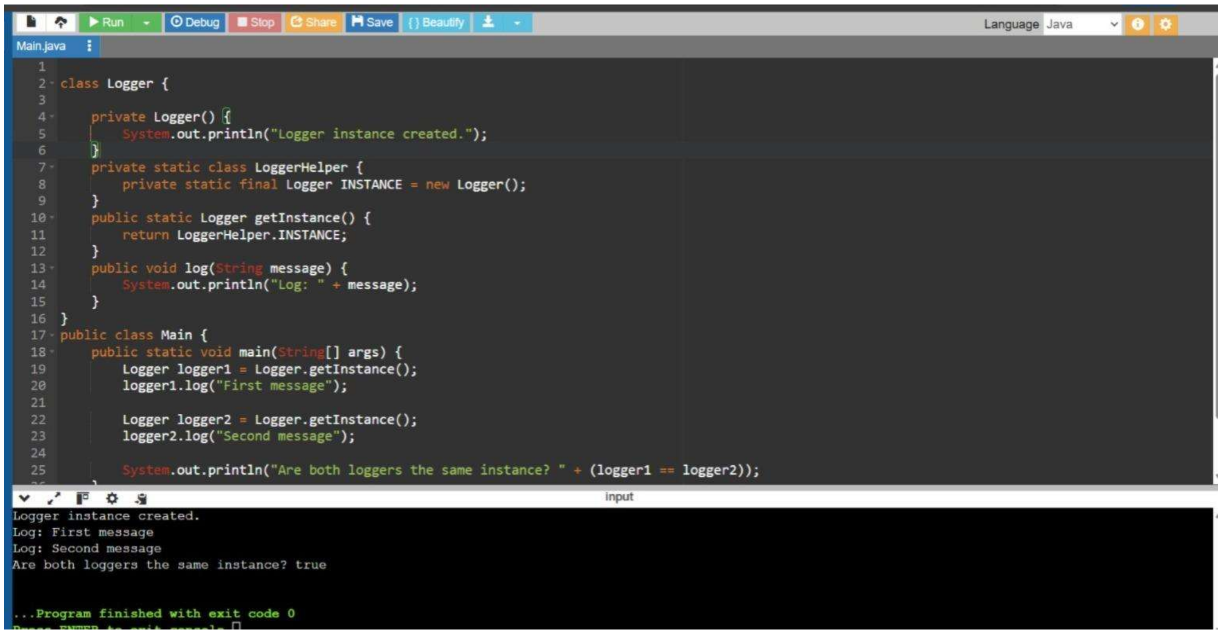
## Scenario:

You need to ensure that a logging utility class in your application has only one instance throughout the application lifecycle to ensure consistent logging.

## Code:

```
class Logger {

   private Logger() {
      System.out.println("Logger instance created.");
   }
   private static class LoggerHelper {
      private static final Logger INSTANCE = new Logger();
   }
   public static Logger getInstance() {
      return LoggerHelper.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("First message");

      Logger logger2 = Logger.getInstance();
      logger2.log("Second message");

      System.out.println("Are both loggers the same instance? " + (logger1 == logger2));
   }
}
```

**Output:**


```java

class Logger {

    private Logger() {
        System.out.println("Logger instance created.");
    }
    private static class LoggerHelper {
        private static final Logger INSTANCE = new Logger();
    }
    public static Logger getInstance() {
        return LoggerHelper.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("First message");

        Logger logger2 = Logger.getInstance();
        logger2.log("Second message");

        System.out.println("Are both loggers the same instance? " + (logger1 == logger2));
```

```
Logger instance created.
Log: First message
Log: Second message
Are both loggers the same instance? true


...Program finished with exit code 0
```

Exercise 2: Implementing the Factory Method Pattern

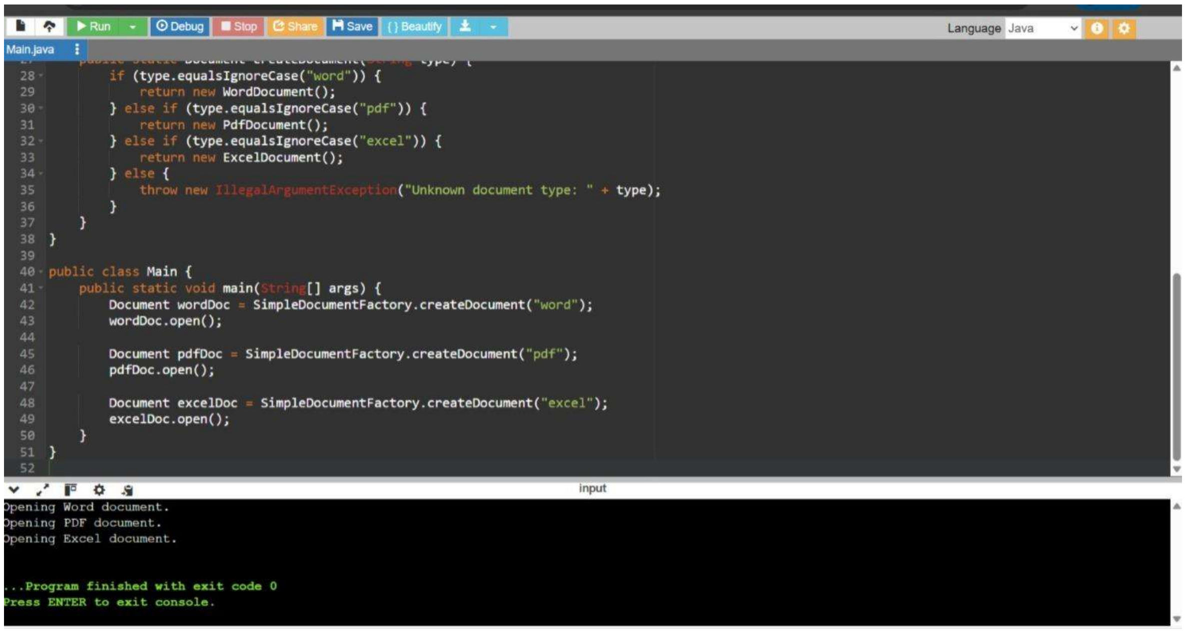
**Scenario:**

You are developing a document management system that needs to create different types of documents (e.g., Word, PDF, Excel). Use the Factory Method Pattern to achieve this.

**Code:**

```java
interface Document {
    void open();
}

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

class PdfDocument implements Document {
    @Override
    public void open() {
        System.out.println("Opening PDF document.");
    }
}

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

class SimpleDocumentFactory {
    public static 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();
        } else {
            throw new IllegalArgumentException("Unknown document type: "
+ type);
        }
    }
}

public class Main {
    public static void main(String[] args) {
        Document wordDoc =
SimpleDocumentFactory.createDocument("word");
        wordDoc.open();

        Document pdfDoc = SimpleDocumentFactory.createDocument("pdf");
        pdfDoc.open();

        Document excelDoc =
SimpleDocumentFactory.createDocument("excel");
        excelDoc.open();
    }
}
```

**Output:**



```java
        if (type.equalsIgnoreCase("word")) {
            return new WordDocument();
        } else if (type.equalsIgnoreCase("pdf")) {
            return new PdfDocument();
        } else if (type.equalsIgnoreCase("excel")) {
            return new ExcelDocument();
        } else {
            throw new IllegalArgumentException("Unknown document type: " + type);
        }
    }
}

public class Main {
    public static void main(String[] args) {
        Document wordDoc = SimpleDocumentFactory.createDocument("word");
        wordDoc.open();

        Document pdfDoc = SimpleDocumentFactory.createDocument("pdf");
        pdfDoc.open();

        Document excelDoc = SimpleDocumentFactory.createDocument("excel");
        excelDoc.open();
    }
}
```

```
Opening Word document.
Opening PDF document.
Opening Excel document.


...Program finished with exit code 0
Press ENTER to exit console.
```