Name:-Ritesh Kumar Sahoo

GitHub Link:- <https://github.com/Riteshksahoo/CognizantDotNet>CognizantDotNet

**-----------------------------------------------------------------------------------------**

**Design Patterns and Principles**

**Exercise 1: Implementing the Singleton Pattern**

**Logger.java:-**

public class Logger {

private static Logger singleInstance;

private Logger() {

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

}

public static Logger getInstance() {

if (singleInstance == null) {

singleInstance = new Logger();

}

return singleInstance;

}

public void log(String message) {

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

}

}

**Main.java :-**

public class Main {

public static void main(String[] args) {

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

System.out.println("Wait....");

Logger logger1 = Logger.getInstance();

Logger logger2 = Logger.getInstance();

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

logger2.log("Continuing the logging....");

if (logger1 == logger2) {

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

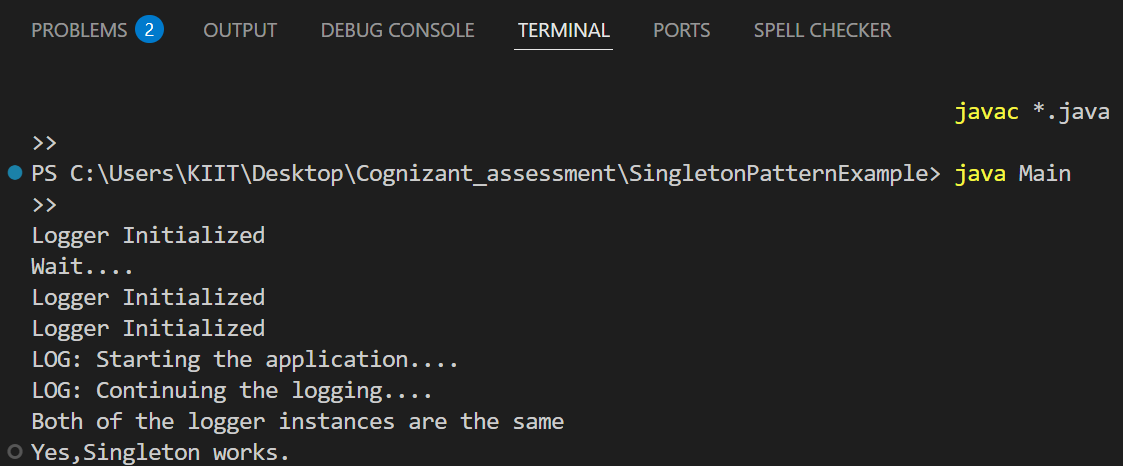
System.err.println("Yes,Singleton works.");

} else {

System.out.println("Logger of the instances are different (Singleton failed).");

}

}

}  
**OUTPUT:-**

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

**Document.java:-**

package FactoryMethodPatternExample;

public interface Document {

void open();

}

**DocumentFactory.java**

package FactoryMethodPatternExample;

public abstract class DocumentFactory {

public abstract Document createDocument();

}  
**ExcelDocument.java**

package FactoryMethodPatternExample;

public class ExcelDocument implements Document {

@Override

public void open() {

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

}

}  
**ExcelFactory.java**package FactoryMethodPatternExample;

public class ExcelFactory extends DocumentFactory {

@Override

public Document createDocument() {

return new ExcelDocument();

}

}

**Main.java**package FactoryMethodPatternExample;

public class Main {

public static void main(String[] args) {

DocumentFactory wordFactory = new WordFactory();

Document wordDoc = wordFactory.createDocument();

wordDoc.open();

DocumentFactory pdfFactory = new PdfFactory();

Document pdfDoc = pdfFactory.createDocument();

pdfDoc.open();

DocumentFactory excelFactory = new ExcelFactory();

Document excelDoc = excelFactory.createDocument();

excelDoc.open();

}

}  
**PdfDocument.java**package FactoryMethodPatternExample;

public class PdfDocument implements Document {

@Override

public void open() {

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

}

}

**PdfFactory.java**package FactoryMethodPatternExample;

public class PdfFactory extends DocumentFactory {

@Override

public Document createDocument()

{

return new PdfDocument();

}

}  
**WordDocument.java**package FactoryMethodPatternExample;

public class WordDocument implements Document {

@Override

public void open() {

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

}

}  
**WordFactory.java**package FactoryMethodPatternExample;

public class WordFactory extends DocumentFactory {

@Override

public Document createDocument() {

return new WordDocument();

}

}  
**OUTUT**