**Exercise 1: Implementing the Singleton Pattern**

**Logger.java**

public class Logger {

private Logger(){

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

}

public static class Singleton{

private static final Logger INSTANTCE = new Logger();

}

public static Logger getInstance(){

return Singleton.INSTANTCE;

}

}

**Main.java**

import java.util.\*;

public class Main {

public static void main(String[] args) {

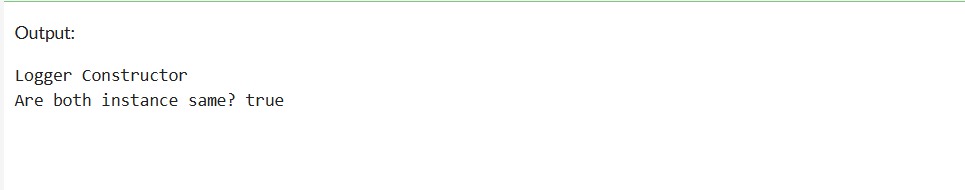
Logger ob1= Logger.getInstance();

Logger ob2=Logger.getInstance();

System.out.println("Are both instance same? "+(ob1==ob2));

}

}



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

**Document.java**

public 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");

}

}

**DocumentFactory.java**

public 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");

}

}

**Main.java**

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();

}

}

