**Exercise 1: Implementing the Singleton Pattern**

**CODE:**

package week1.DPP;

class Logger{

    private static Logger instance;

    private Logger(){

        System.out.println("Logger instance created!");

    }

    public static Logger getInstance(){

        if(instance == null){

            instance = new Logger();

        }

        return instance;

    }

    public void log(String msg){

        System.out.println("Log: " + msg);

    }

}

public class SingletonPatternExample{

    public static void main(String[] args){

        Logger logger1 = Logger.getInstance();

        Logger logger2 = Logger.getInstance();

        logger1.log("Start the application");

        logger2.log("continue the application");

        if(logger1 == logger2){

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

        }

        else{

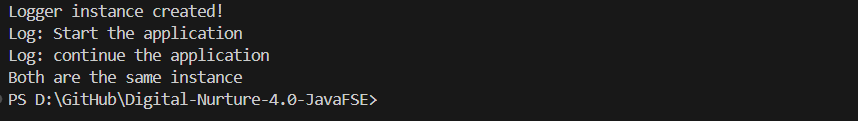
            System.out.println("Both are different instance");

        }

    }

}

**OUTPUT:**

****

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

**CODE:**

package week1.DPP;

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");

    }

}

abstract class DocumentFactory{

    public abstract Document createDocument();

}

class WordDocumentFactory extends DocumentFactory{

    @Override

    public Document createDocument(){

        return new WordDocument();

    }

}

class PdfDocumentFactory extends DocumentFactory{

    @Override

    public Document createDocument(){

        return new PdfDocument();

    }

}

class ExcelDocumentFactory extends DocumentFactory{

    @Override

    public Document createDocument(){

        return new ExcelDocument();

    }

}

public class FactoryMethodPatternExample {

    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:**

