**SINGLETON PATTERN EXAMPLE**

**CODE:**

**Logger.java**

package SingletonPatternExample;

public class Logger{

    private static Logger log\_instance;

    private Logger(){

    }

    public static Logger getInstance() {

        if (log\_instance == null) {

            log\_instance = new Logger();

        }

        return log\_instance;

    }

    public void log(String message) {

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

    }

}

**LoggerTest.java**

package SingletonPatternExample;

public class LoggerTest {

    public static void main(String[] args) {

        Logger logger1 = Logger.getInstance();

        Logger logger2 = Logger.getInstance();

        logger1.log("First log message.");

        logger2.log("Second log message.");

        if (logger1 == logger2) {

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

        } else {

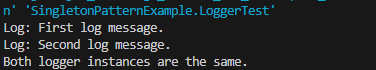
            System.out.println("Different logger instances exist.");

        }

    }

}

**OUTPUT:**



**FACTORY METHOD PATTERN EXAMPLE**

**CODE:**

**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();

    }

}

**FactoryTest.java**

package FactoryMethodPatternExample;

public class FactoryTest {

    public static void main(String[] args) {

        // Word Document

        DocumentFactory wordFactory = new WordFactory();

        Document wordDoc = wordFactory.createDocument();

        wordDoc.open();

        // PDF Document

        DocumentFactory pdfFactory = new PdfFactory();

        Document pdfDoc = pdfFactory.createDocument();

        pdfDoc.open();

        // Excel Document

        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();

    }

}

**OUTPUT:**

