**SingletonPatternExample**.  
**Steps:**

1. **Create a New Java Project:**

○ Create a new Java project named **SingletonPatternExample**.

2. **Define a Singleton Class:**

○ Create a class named Logger that has a private static instance of itself.

○ Ensure the constructor of Logger is private.

○ Provide a public static method to get the instance of the Logger class.

3. **Implement the Singleton Pattern:**

○ Write code to ensure that the Logger class follows the Singleton design pattern.

4. **Test the Singleton Implementation:**

○ Create a test class to verify that only one instance of Logger is created and used across the application.

**SOURCE CODE :**

class single\_ton {

private static single\_ton *instance*;

private single\_ton(){

}

public static single\_ton getInstance() {

if(*instance*==null){

*instance*=new single\_ton();

}

return *instance*;

}

public void statement(String s){

System.*out*.println("instance created"+" "+s);

}

}

public class Singleton {

public static void main(String[] args) {

single\_ton log1= single\_ton.*getInstance*();

log1.statement("Person 1");

single\_ton log2= single\_ton.*getInstance*();

log2.statement("Person 2");

if(log2==log1){

System.*out*.println("Have equal instance");

}

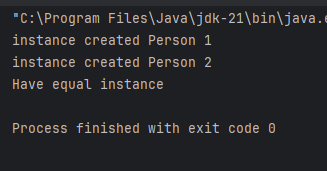
else{

System.*out*.println("Have no equal instance");

}

}

}

**OUTPUT :**   
  


### **FactoryMethodPatternDemo**

**Steps:**

1. **Create a New Java Project:** ○ Start a new Java project and name it **FactoryMethodPatternDemo**.
2. **Define Document Interface:** ○ Create an interface named **Doc** that declares a method called openDoc() to simulate opening a document.
3. **Create Concrete Document Classes:** ○ Implement concrete classes **WordDoc**, **PdfDoc**, and **ExcelDoc** that implement the Doc interface and override the openDoc() method with their respective implementations.
4. **Implement the Factory Method Pattern:** ○ Create an abstract class called **DocFactory** with an abstract method createDoc().  
    ○ Define concrete factory classes **WordDocFactory**, **PdfDocFactory**, and **ExcelDocFactory** that extend DocFactory and implement the createDoc() method to return the corresponding document type.

### **Test the Factory Method Implementation:** ○ Create a main test class named **FactoryMethodPatternDemo**. ○ In this class, use the concrete factories to create and open different types of documents (WordDoc, PdfDoc, ExcelDoc) using the factory method pattern.

**SOURCE CODE :**

**public class Factory\_Method\_Pattern {**

**interface Doc {**

**void openDoc();**

**}**

**static class WordDoc implements Doc {**

**public void openDoc() {**

**System.*out*.println("Word document is opened.");**

**}**

**}**

**static class PdfDoc implements Doc {**

**public void openDoc() {**

**System.*out*.println("PDF document is opened.");**

**}**

**}**

**static class ExcelDoc implements Doc {**

**public void openDoc() {**

**System.*out*.println("Excel document is opened.");**

**}**

**}**

**static abstract class DocFactory {**

**public abstract Doc createDoc();**

**}**

**static class WordDocFactory extends DocFactory {**

**public Doc createDoc() {**

**return new WordDoc();**

**}**

**}**

**static class PdfDocFactory extends DocFactory {**

**public Doc createDoc() {**

**return new PdfDoc();**

**}**

**}**

**static class ExcelDocFactory extends DocFactory {**

**public Doc createDoc() {**

**return new ExcelDoc();**

**}**

**}**

**public static class FactoryMethodPatternDemo {**

**public static void main(String[] args) {**

**Factory\_Method\_Pattern.DocFactory wordCreator = new Factory\_Method\_Pattern.WordDocFactory();**

**Factory\_Method\_Pattern.Doc wordFile = wordCreator.createDoc();**

**wordFile.openDoc();**

**Factory\_Method\_Pattern.DocFactory pdfCreator = new Factory\_Method\_Pattern.PdfDocFactory();**

**Factory\_Method\_Pattern.Doc pdfFile = pdfCreator.createDoc();**

**pdfFile.openDoc();**

**Factory\_Method\_Pattern.DocFactory excelCreator = new Factory\_Method\_Pattern.ExcelDocFactory();**

**Factory\_Method\_Pattern.Doc excelFile = excelCreator.createDoc();**

**excelFile.openDoc();**

**}**

**}**

**}**

**OUTPUT :**

