**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.

Code:

package Week\_1\_Algorithm\_and\_Data\_Structures;

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("user1");

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

log2.statement("user2");

if(log2==log1){

System.*out*.println("Both instance are equal");

}

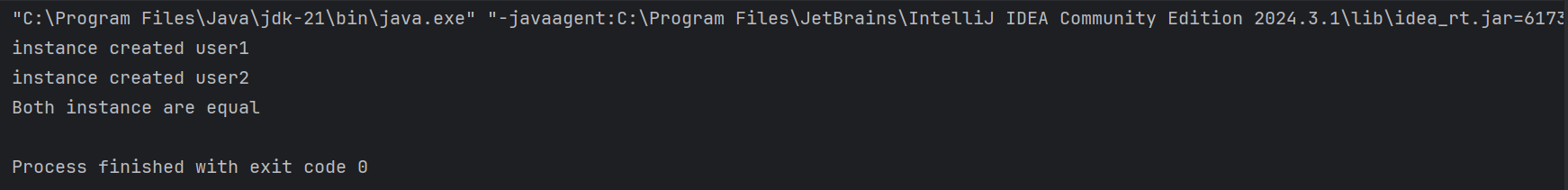
else{

System.*out*.println("Both instance are not equal");

}

}

}

Output:  


**FactoryMethodPatternExample**

**Steps:**

1. **Create a New Java Project:**
   * Create a new Java project named **FactoryMethodPatternExample**.
2. **Define Document Classes:**
   * Create interfaces or abstract classes for different document types such as **WordDocument**, **PdfDocument**, and **ExcelDocument**.
3. **Create Concrete Document Classes:**
   * Implement concrete classes for each document type that implements or extends the above interfaces or abstract classes.
4. **Implement the Factory Method:**
   * Create an abstract class **DocumentFactory** with a method **createDocument()**.
   * Create concrete factory classes for each document type that extends DocumentFactory and implements the **createDocument()** method.
5. **Test the Factory Method Implementation:**
   * Create a test class to demonstrate the creation of different document types using the factory method.  
       
     class DocumentFactoryExample {
   * interface Document {
   * void open();
   * }
   * static class WordDocument implements Document {
   * public void open() {
   * System.*out*.println("Opening a Word document.");
   * }
   * }
   * static class PdfDocument implements Document {
   * public void open() {
   * System.*out*.println("Opening a PDF document.");
   * }
   * }
   * static class ExcelDocument implements Document {
   * public void open() {
   * System.*out*.println("Opening an Excel document.");
   * }
   * }
   * static abstract class DocumentFactory {
   * public abstract Document createDocument();
   * }
   * static class WordDocumentFactory extends DocumentFactory {
   * public Document createDocument() {
   * return new WordDocument();
   * }
   * }
   * static class PdfDocumentFactory extends DocumentFactory {
   * public Document createDocument() {
   * return new PdfDocument();
   * }
   * }
   * static class ExcelDocumentFactory extends DocumentFactory {
   * @Override
   * public Document createDocument() {
   * return new ExcelDocument();
   * }
   * }
   * }
   * public class Week\_1\_Data\_Structures\_and\_Algorithms {
   * public static void main(String[] args) {
   * DocumentFactoryExample.DocumentFactory wordFactory = new DocumentFactoryExample.WordDocumentFactory();
   * DocumentFactoryExample.Document wordDoc = wordFactory.createDocument();
   * wordDoc.open();
   * DocumentFactoryExample.DocumentFactory pdfFactory = new DocumentFactoryExample.PdfDocumentFactory();
   * DocumentFactoryExample.Document pdfDoc = pdfFactory.createDocument();
   * pdfDoc.open();
   * DocumentFactoryExample.DocumentFactory excelFactory = new DocumentFactoryExample.ExcelDocumentFactory();
   * DocumentFactoryExample.Document excelDoc = excelFactory.createDocument();
   * excelDoc.open();
   * }
   * }
   * Output:  
     