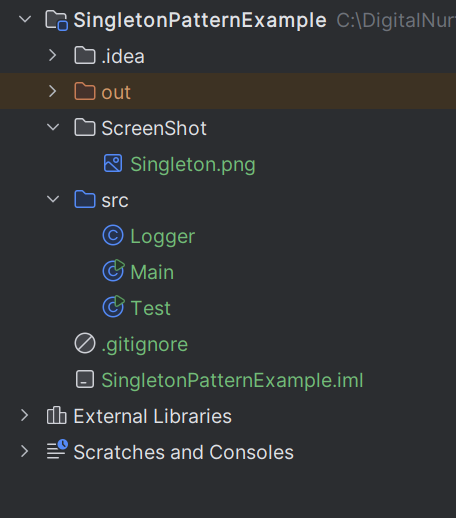
**Exercise 1: Implementing the Singleton Pattern**

**Logger.java**

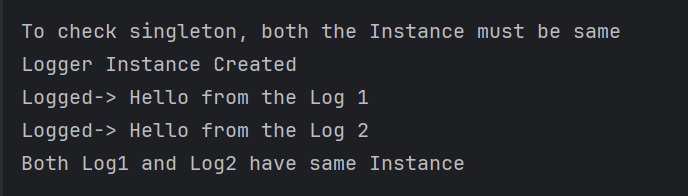
public class Logger {  
 private static Logger *instance*;  
  
 private Logger(){  
 System.*out*.println("Logger Instance Created ");  
 }  
  
 public static Logger getInstance(){  
 if(*instance*==null){  
 *instance*=new Logger();  
 }  
 return *instance*;  
 }  
  
 public void Log(String s){  
 System.*out*.println("Logged-> "+s);  
 }  
  
}

**Test.java**

public class Test {  
 public static void main(String[] args) {  
 System.*out*.println();  
 System.*out*.println("To check singleton, both the Instance must be same");  
 Logger l1= Logger.*getInstance*();  
 Logger l2= Logger.*getInstance*();  
 l1.Log("Hello from the Log 1 ");  
 l2.Log("Hello from the Log 2 ");  
 if(l1==l2){  
 System.*out*.println("Both Log1 and Log2 have same Instance ");  
 }  
 else{  
 System.*out*.println("L1 and L2 are different Instance ");  
 }  
  
 }  
}



**Output**



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

**Document.java**

public abstract class Document {  
 abstract String DocType();  
  
 abstract String Extention();  
}  
class WordDocument extends Document{  
  
 @Override  
 String DocType() {  
 return "Word";  
 }  
  
 @Override  
 String Extention() {  
 return ".docx";  
 }  
}  
  
class PdfDocument extends Document{  
  
 @Override  
 String DocType() {  
 return "Pdf";  
 }  
  
 @Override  
 String Extention() {  
 return ".pdf";  
 }  
}  
  
class ExcelDocument extends Document{  
  
 @Override  
 String DocType() {  
 return "Excel";  
 }  
  
 @Override  
 String Extention() {  
 return ".xlsx";  
 }  
}

**DocumentFactory.java**

public abstract class DocumentFactory {  
 abstract Document createDocument();  
}  
  
class wordDocumentFactory extends DocumentFactory{  
 @Override  
 Document createDocument(){  
 return new WordDocument();  
 }  
}  
  
class PdfDocumentFactory extends DocumentFactory{  
 @Override  
 Document createDocument(){  
 return new PdfDocument();  
 }  
}

class ExcelDocumentFactory extends DocumentFactory {  
 @Override  
 Document createDocument() {  
 return new ExcelDocument();  
 }  
}

**FactoryMethodTest.java**

public class FactoryMethodTest {  
 public static void main(String[] args) {  
 DocumentFactory wordFactory = new wordDocumentFactory();  
 Document wordDoc = wordFactory.createDocument();  
 System.*out*.println("Type: " + wordDoc.DocType() + ", Extension: " + wordDoc.Extention());  
  
 DocumentFactory pdfFactory = new PdfDocumentFactory();  
 Document pdfDoc = pdfFactory.createDocument();  
 System.*out*.println("Type: " + pdfDoc.DocType() + ", Extension: " + pdfDoc.Extention());  
  
 DocumentFactory excelFactory = new ExcelDocumentFactory();  
 Document excelDoc = excelFactory.createDocument();  
 System.*out*.println("Type: " + excelDoc.DocType() + ", Extension: " + excelDoc.Extention());  
 }  
}

**Output**

