A close-up of a logo

AI-generated content may be incorrect.Digital Nurture **4.0 – Deep Skilling (Java FSE)**

Name : Aswath Kannan M  
Superset ID : 6420716 (Week 1)  
  
**Design Pattern and Principles  
Qn 1:** Implementing the Singleton Pattern

Logger. Java

// Source code

package com.singleton.example;

public class Logger {

   private static Logger instance;

 private Logger() {

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

   }

 public static Logger getInstance() {

      if (instance == null) {

         Class var0 = Logger.class;

         synchronized(Logger.class) {

            if (instance == null) {

               instance = new Logger();

            }  } }

return instance;

   }

 public void log(String var1) {

      System.out.println("Log: " + var1);

   } }

SingletonTest.java  
  
// Source code

package com.singleton.example;

public class SingletonTest {

   public SingletonTest() {

   }

public static void main(String[] var0) {

      Logger var1 = Logger.getInstance();

      var1.log("This is the first log message.");

      Logger var2 = Logger.getInstance();

      var2.log("This is the second log message.");

      if (var1 == var2) {

         System.out.println("\nBoth logger1 and logger2 refer to the same instance.");

      } else {

         System.out.println("\nError: Multiple instances of Logger created.");

      }

  System.out.println("Hashcode of logger1: " + var1.hashCode());

    System.out.println("Hashcode of logger2: " + var2.hashCode());

   }

}

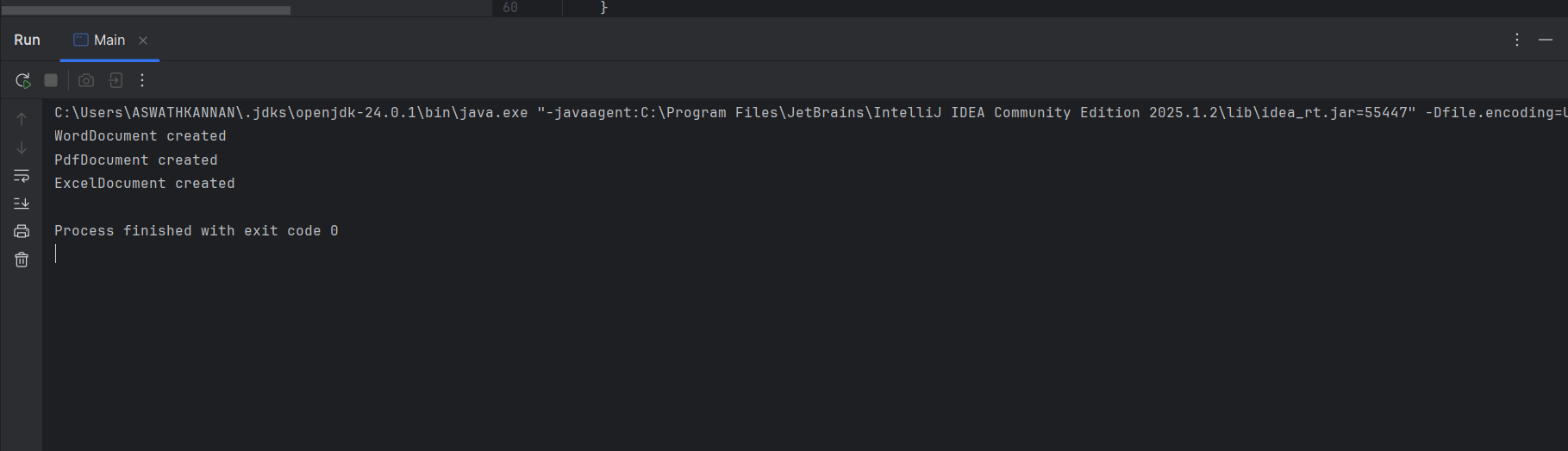
A screen shot of a computer program

AI-generated content may be incorrect.**Output**

**Qn 2 :** Implementing the Factory Method Pattern  
Main.Java

// All classes are in the default package  
interface document{  
 String create();  
}  
  
class WordDocument implements document{  
 public String create(){  
 return "WordDocument created";  
 }  
}  
  
class PdfDocument implements document{  
 public String create(){  
 return "PdfDocument created";  
 }  
}  
  
class ExcelDocument implements document{  
 public String create(){  
 return "ExcelDocument created";  
 }  
}  
  
abstract class DocumentFactory {  
 abstract document createDocument();  
}  
  
class WordDocumentFactory extends DocumentFactory{  
 public document createDocument(){  
 return new WordDocument();  
 }  
}  
  
class PdfDocumentFactory extends DocumentFactory{  
 public document createDocument(){  
 return new PdfDocument();  
 }  
}  
  
class ExcelDocumentFactory extends DocumentFactory{  
 public document createDocument(){  
 return new ExcelDocument();  
 }  
}  
  
public class Main { // Only this class is public  
 public static void main(String[] args) {  
 // Using distinct variable names (factory1, doc1, etc.) to avoid re-declaration errors  
 DocumentFactory factory1 = new WordDocumentFactory();  
 document doc1 = factory1.createDocument();  
 System.*out*.println(doc1.create());  
  
 DocumentFactory factory2 = new PdfDocumentFactory();  
 document doc2 = factory2.createDocument();  
 System.*out*.println(doc2.create());  
  
 DocumentFactory factory3 = new ExcelDocumentFactory();  
 document doc3 = factory3.createDocument();  
 System.*out*.println(doc3.create());  
 }  
}  
  
**Output**   
  
A computer screen shot of a program

AI-generated content may be incorrect.

****