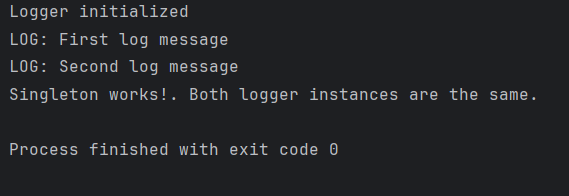
**Exercise 1: Implementing the Singleton Pattern**

**Logger.java**

package singleton;  
  
public class Logger {  
 private static Logger *instance*;  
 private Logger() {  
 System.*out*.println("Logger initialized");  
 }  
 public static Logger getInstance() {  
 if (*instance* == null) {  
 *instance* = new Logger();  
 }  
 return *instance*;  
 }  
 public void log(String message) {  
 System.*out*.println("LOG: " + message);  
 }  
}

**SingletonTest.java**

package singleton;  
  
public class SingletonTest {  
 public static void main(String[] args) {  
 Logger logger1 = Logger.*getInstance*();  
 logger1.log("First log message");  
  
 Logger logger2 = Logger.*getInstance*();  
 logger2.log("Second log message");  
  
 if (logger1 == logger2) {  
 System.*out*.println("Singleton works!. Both logger instances are the same.");  
 } else {  
 System.*out*.println("Singleton failed! Logger instances are different.");  
 }  
 }  
}

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

**Document.java**

package factory;  
  
public interface Document {  
 void open();  
}

**WordDocument.java**

package factory;  
  
public class WordDocument implements Document {  
 @Override  
 public void open() {  
 System.*out*.println("Opening Word document...");  
 }  
}

**ExcelDocument.java**

package factory;  
  
public class ExcelDocument implements Document {  
 @Override  
 public void open() {  
 System.*out*.println("Opening Excel document...");  
 }  
}

**PdfDocument.java**

package factory;  
  
public class PdfDocument implements Document {  
 @Override  
 public void open() {  
 System.*out*.println("Opening PDF document...");  
 }  
}

**WordDocumentFactory.java**

package factory;  
  
public class WordDocumentFactory extends DocumentFactory {  
 @Override  
 public Document createDocument() {  
 return new WordDocument();  
 }  
}

**ExcelDocumentFactory.java**

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

**PdfDocumentFactory.java**

package factory;  
  
public class PdfDocumentFactory extends DocumentFactory {  
 @Override  
 public Document createDocument() {  
 return new PdfDocument();  
 }  
}

**DocumentFactory.java**

package factory;  
  
public abstract class DocumentFactory {  
 public abstract Document createDocument();  
}

**FactoryProvider.java**

package factory;  
  
public class FactoryProvider {  
  
 public static DocumentFactory getFactory(String type) {  
 if (type == null) return null;  
  
 switch (type.toLowerCase()) {  
 case "word":  
 return new WordDocumentFactory();  
 case "pdf":  
 return new PdfDocumentFactory();  
 case "excel":  
 return new ExcelDocumentFactory();  
 default:  
 System.*out*.println("Unknown document type: " + type);  
 return null;  
 }  
 }  
}

**FactoryTest.java**

package factory;  
  
import java.util.Scanner;  
  
public class FactoryTest {  
 public static void main(String[] args) {  
 Scanner scanner = new Scanner(System.*in*);  
  
 System.*out*.println("Enter document type to create (word / pdf / excel): ");  
 String input = scanner.nextLine();  
  
 DocumentFactory factory = FactoryProvider.*getFactory*(input);  
  
 if (factory != null) {  
 Document doc = factory.createDocument();  
 doc.open();  
 } else {  
 System.*out*.println("Failed to create document. Please enter a valid type.");  
 }  
  
 scanner.close();  
 }  
}

