**1.Implementing the Singleton Pattern**

To ensure that a logging utility class in my application has only one instance throughout the application lifecycle to ensure consistent logging.

**Logger.java**package com.example.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);

}

**BankingApp.java**

package com.example.singleton;

public class BankingApp {

public static void main(String[] args) {

Logger logger1 = Logger.*getInstance*();

logger1.log("Account created for user: Durga Thangaraj");

logger1.log("Deposited ₹10,000 into account number 985783456790");

logger1.log("Withdrawn ₹3,000 from account number 985783456790");

Logger logger2 = Logger.*getInstance*();

logger2.log("Checked balance for account number 985783456790");

if (logger1 == logger2) {

System.*out*.println("Only one instance of Logger is used. Singleton works!");

} else {

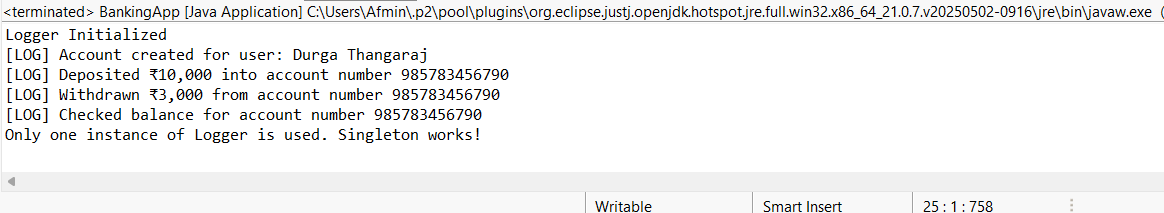
System.*out*.println("Different Logger instances found. Singleton failed!");

}

}

}

**Output**



**2**  **Implementing the Factory Method Pattern**

To develop a document management system such as pdf,word,xlsx.

**Document.java**

package com.documents;

public abstract class DocumentFactory {

public abstract Document createDocument();

}  
**DocumentFactory.java**

package com.documents;

public abstract class DocumentFactory {

public abstract Document createDocument();

}

DocumentFactoryTest.java

package com.documents;

public class DocumentFactoryTest {

public static void main(String[] args) {

DocumentFactory wordFactory = new WordDocumentFactory();

Document wordDoc = wordFactory.createDocument

wordDoc.open();

DocumentFactory pdfFactory = new PdfDocumentFactory();

Document pdfDoc = pdfFactory.createDocument();

pdfDoc.open();

DocumentFactory excelFactory = new ExcelDocumentFactory();

Document excelDoc = excelFactory.createDocument();

excelDoc.open();

}

}  
**ExcelDocument.java**

package com.documents;

import java.awt.Desktop;

import java.io.File;

import java.io.IOException;

public class ExcelDocument implements Document {

@Override

public void open() {

System.***out***.println("Opening an Excel document...");

**try** {

File file = new File("C:\\Users\\Durga \\Downloads\\sample.xlsx");

**if** (file.exists()) {

Desktop.*getDesktop*().open(file);

} **else** {

System.***out***.println("File not found.");

}

} **catch** (IOException e) {

System.***out***.println("Error opening the file: " + e.getMessage());

}}}

**ExcelDocument.java**

package com.documents;

**public** **class** ExcelDocumentFactory **extends** DocumentFactory {

@Override

**public** Document createDocument() {

**return** **new** ExcelDocument();

}

}

**PdfDocument.java**package com.documents;

import java.awt.Desktop;

import java.io.File;

import java.io.IOException;

public class PdfDocument implements Document {

@Override

public void open() {

System.*out*.println("Opening a PDF document...");

try {

File file = new File("C:\\Users\\Durga\\Downloads\\sample.pdf");

if (file.exists()) {

Desktop.*getDesktop*().open(file);

} else {

System.*out*.println("File not found.");

}

} catch (IOException e) {

System.*out*.println("Error opening the file: " + e.getMessage());

}

}}  
**PdfDocumentFactory.java**

**package** com.documents;

**public** **class** PdfDocumentFactory **extends** DocumentFactory {

@Override

**public** Document createDocument() {

**return** **new** PdfDocument();

}

}

**WordDocument.java**

package com.documents;

import java.awt.Desktop;

import java.io.File;

import java.io.IOException;

public class WordDocument implements Document {

@Override

public void open() {

System.*out*.println("Opening a Word document...");

try {

File file = new File("C:\\Users\\Durga \\Downloads\\sample.docx");

if (file.exists()) {

Desktop.*getDesktop*().open(file);

} else {

System.*out*.println("File not found.");

}

} catch (IOException e) {

System.*out*.println("Error opening the file: " + e.getMessage());

}

}

}

**WordDocumentFactory.java**package com.documents;

public class WordDocumentFactory extends DocumentFactory {

@Override

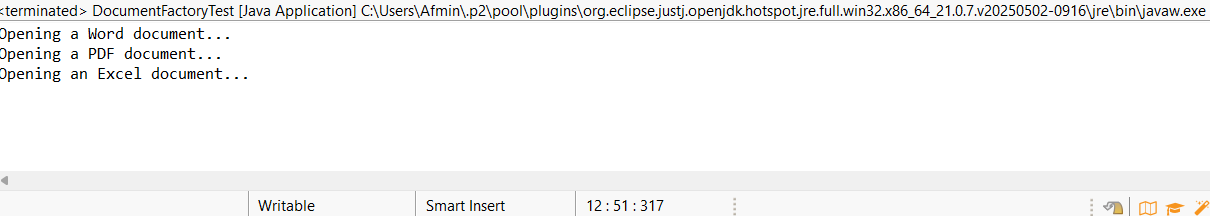
public Document createDocument() {

return new WordDocument();

}

}

**OUTPUT**

****

