**Week-1 Design Patterns and Principles**

1. **Implementing the Singleton Pattern**

**Scenario:**

You need to ensure that a logging utility class in your application has only one instance throughout the application lifecycle to ensure consistent logging.

**Code:**

**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);

}

}

**LoggerTest.java**

**package** singleton;

**public** **class** LoggerTest {

**public** **static** **void** main(String[] args) {

Logger logger1 = Logger.*getInstance*();

logger1.log("This is the first log message.");

Logger logger2 = Logger.*getInstance*();

logger2.log("This is the second log message.");

**if** (logger1 == logger2) {

System.***out***.println("Both logger instances are the same (singleton confirmed).");

}

**else** {

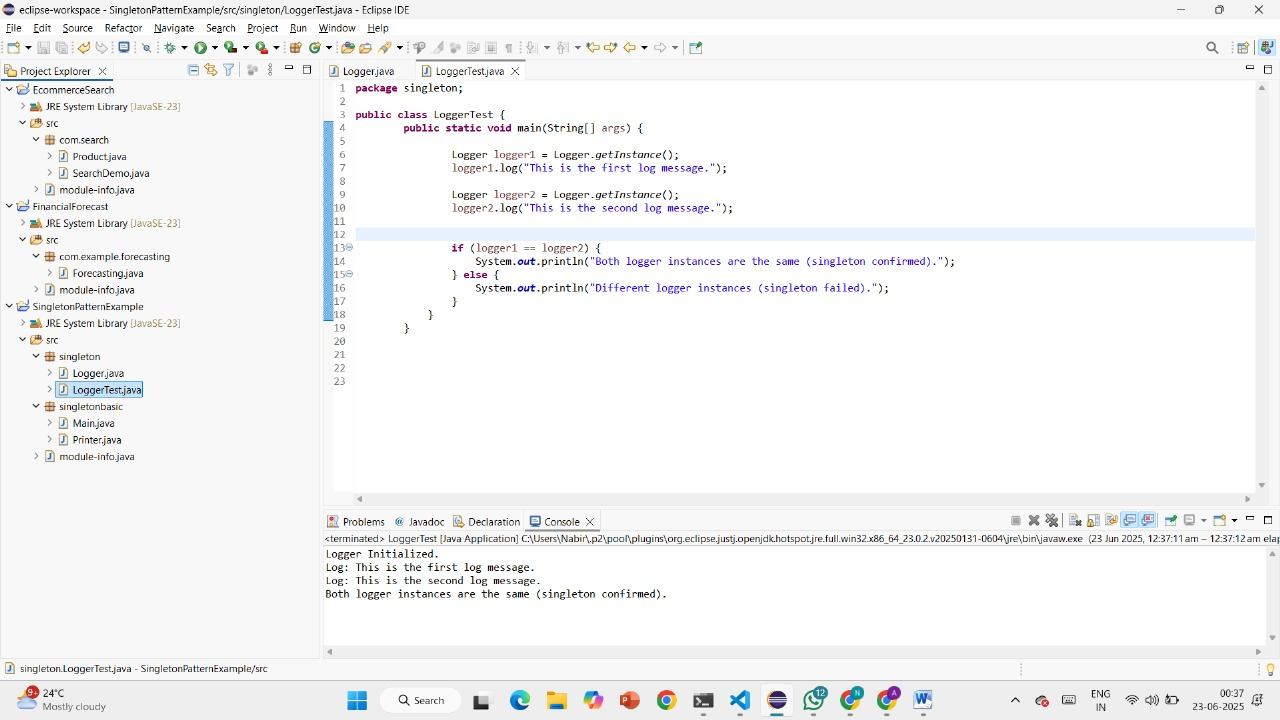
System.***out***.println("Different logger instances (singleton failed).");

}

}

}

**Output:**



1. **Implementing the Factory Method Pattern**

**Scenario:**

You are developing a document management system that needs to create different types of documents (e.g., Word, PDF, Excel). Use the Factory Method Pattern to achieve this.

**Code:**

**Document.java**

**package** factorymethodexample;

**public** **interface** Document {

**void** open();

}

**WordDoc.java**

**package** factorymethodexample;

**public** **class** WordDoc **implements** Document {

**public** **void** open() {

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

}

}

**Pdf.java**

**package** factorymethodexample;

**public** **class** Pdf **implements** Document {

**public** **void** open() {

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

}

}

**Excel.java**

**package** factorymethodexample;

**public** **class** Excel **implements** Document {

**public** **void** open() {

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

}

}

**DocumentFactory.java**

**package** factorymethodexample;

**public** **abstract** **class** DocumentFactory {

**public** **abstract** Document createDocument();

}

**WordDocumentFactory.java**

**package** factorymethodexample;

**public** **class** WordDocumentFactory **extends** DocumentFactory {

**public** Document createDocument() {

**return** **new** WordDoc();

}

}

**PdfDocumentFactory.java**

**package** factorymethodexample;

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

**public** Document createDocument() {

**return** **new** Pdf();

}

}

**ExcelDocumentFactory.java**

**package** factorymethodexample;

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

**public** Document createDocument() {

**return** **new** Excel();

}

}

**Main.java**

**package** factorymethodexample;

**public** **class** Main {

**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();

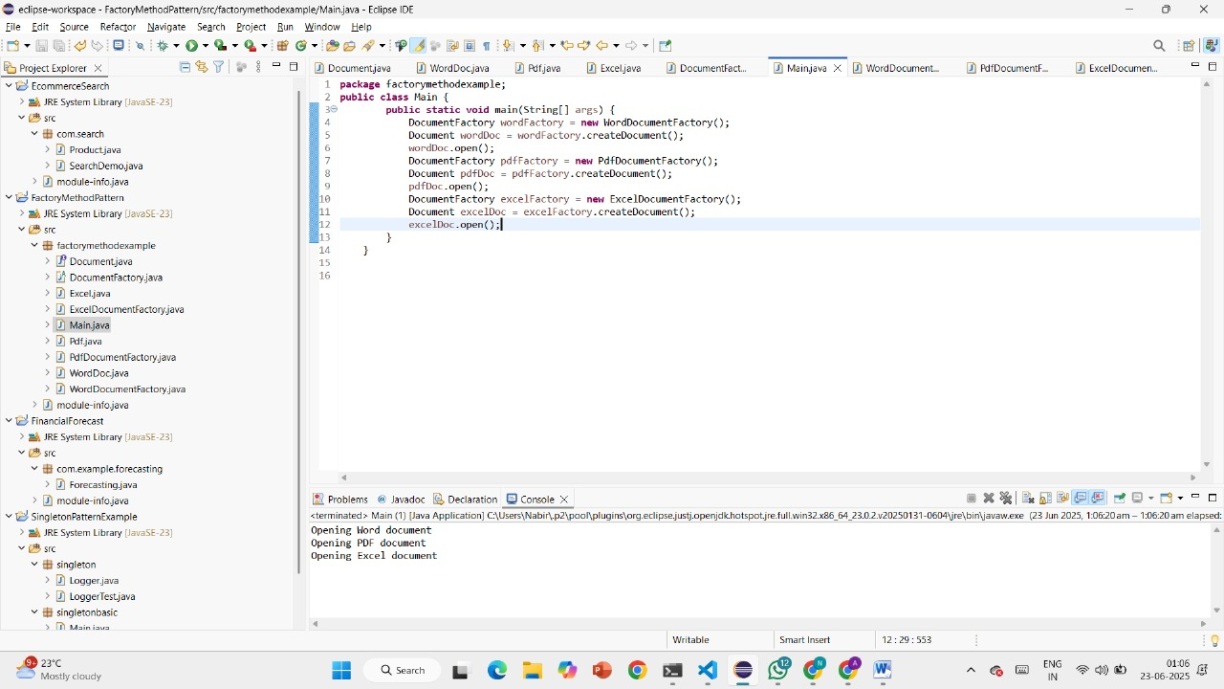
Document excelDoc = excelFactory.createDocument();

excelDoc.open();

}

}

**Output:**

****