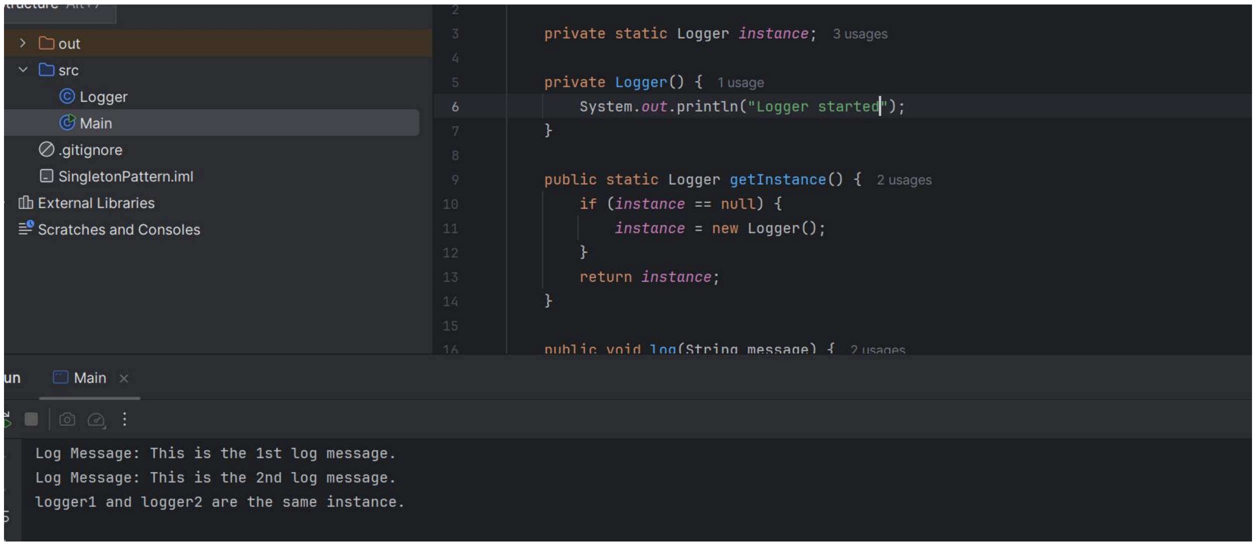
OUTPUT SCREENSHOTS

Week1: Design Patterns And Principles

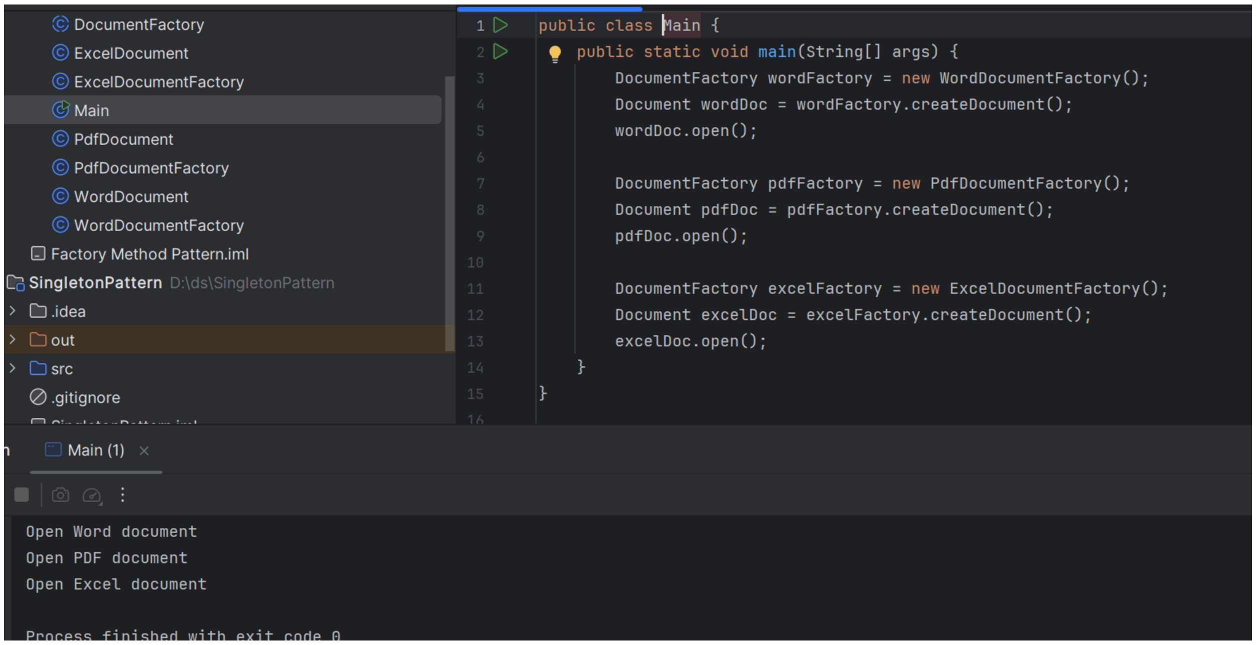
Implementing the Singleton Pattern



```java
    private static Logger instance;  3 usages

    private Logger() {  1 usage
        System.out.println("Logger started");
    }

    public static Logger getInstance() {  2 usages
        if (instance == null) {
            instance = new Logger();
        }
        return instance;
    }

    public void log(String message) {  2 usages
```

```
Log Message: This is the 1st log message.
Log Message: This is the 2nd log message.
logger1 and logger2 are the same instance.
```

2. Implementing the Factory Method Pattern

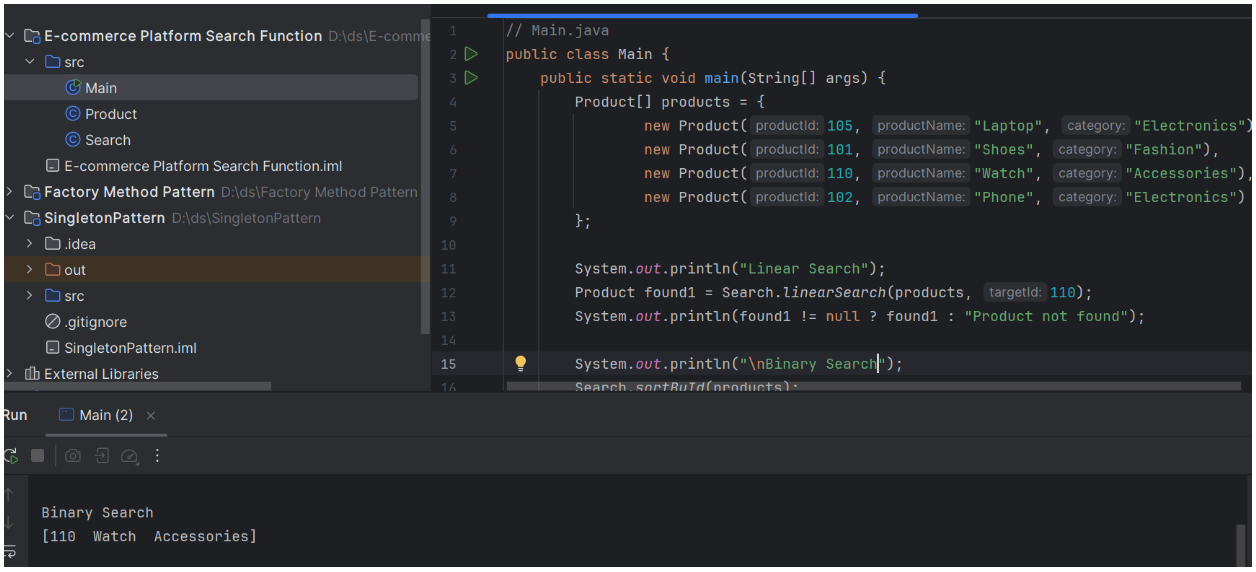


```java
public class Main {
    public static void main(String[] args) {
        DocumentFactory wordFactory = new WordDocumentFactory();
        Document wordDoc = wordFactory.createDocument();
        wordDoc.open();

        DocumentFactory pdfFactory = new PdfDocumentFactory();
        Document pdfDoc = pdfFactory.createDocument();
        pdfDoc.open();

        DocumentFactory excelFactory = new ExcelDocumentFactory();
        Document excelDoc = excelFactory.createDocument();
        excelDoc.open();
    }
}
```

```
Open Word document
Open PDF document
Open Excel document

Process finished with exit code 0
```

Week 1: Data structures and Algorithms

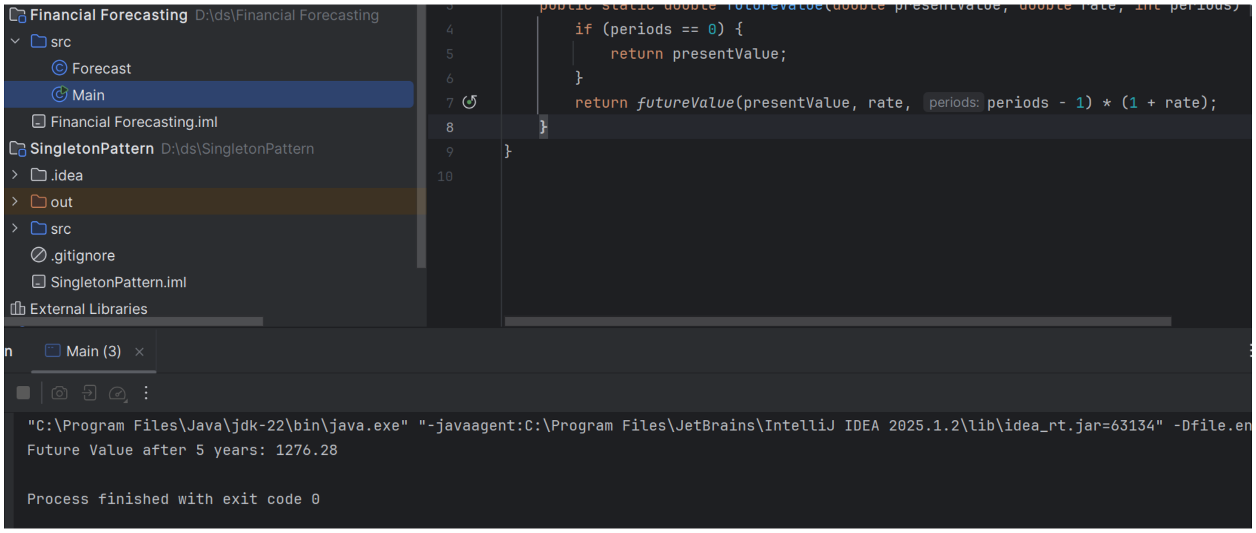
1.E-commerce Platform Search Function


E-commerce Platform Search Function D:\ds\E-comme
src
Main
Product
Search
E-commerce Platform Search Function.iml
Factory Method Pattern D:\ds\Factory Method Pattern
SingletonPattern D:\ds\SingletonPattern
.idea
out
src
.gitignore
SingletonPattern.iml
External Libraries
Run
Main (2)

```java
// Main.java
public class Main {
    public static void main(String[] args) {
        Product[] products = {
                new Product( productId: 105, productName: "Laptop", category: "Electronics")
                new Product( productId: 101, productName: "Shoes", category: "Fashion"),
                new Product( productId: 110, productName: "Watch", category: "Accessories"),
                new Product( productId: 102, productName: "Phone", category: "Electronics")
        };

        System.out.println("Linear Search");
        Product found1 = Search.linearSearch(products, targetId: 110);
        System.out.println(found1 != null ? found1 : "Product not found");

        System.out.println("\nBinary Search");
        Search.sortById(products);
```

```
Binary Search
[110  Watch  Accessories]
```

2. Financial Forecasting


Financial Forecasting D:\ds\Financial Forecasting
src
Forecast
Main
Financial Forecasting.iml
SingletonPattern D:\ds\SingletonPattern
.idea
out
src
.gitignore
SingletonPattern.iml
External Libraries
Main (3)

```java
    public static double futureValue(double presentValue, double rate, int periods)
        if (periods == 0) {
            return presentValue;
        }
        return futureValue(presentValue, rate, periods: periods - 1) * (1 + rate);
    }
}
```

```
"C:\Program Files\Java\jdk-22\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2025.1.2\lib\idea_rt.jar=63134" -Dfile.en
Future Value after 5 years: 1276.28

Process finished with exit code 0
```