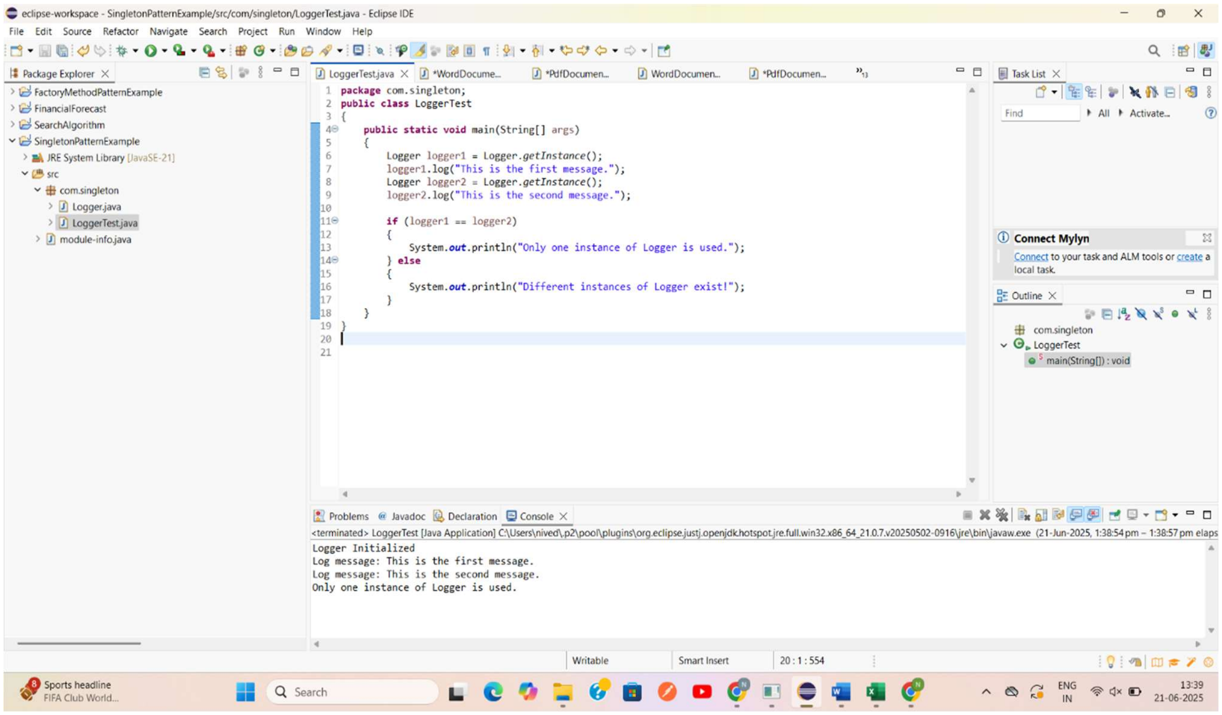
WEAK 1:COGNIZANT HANDS-ON PROBLEMS

TOPIC:1 DESIGN PRICIPLES AND PATTERNS

Exercise 1:Implement the singleton pattern

Output:

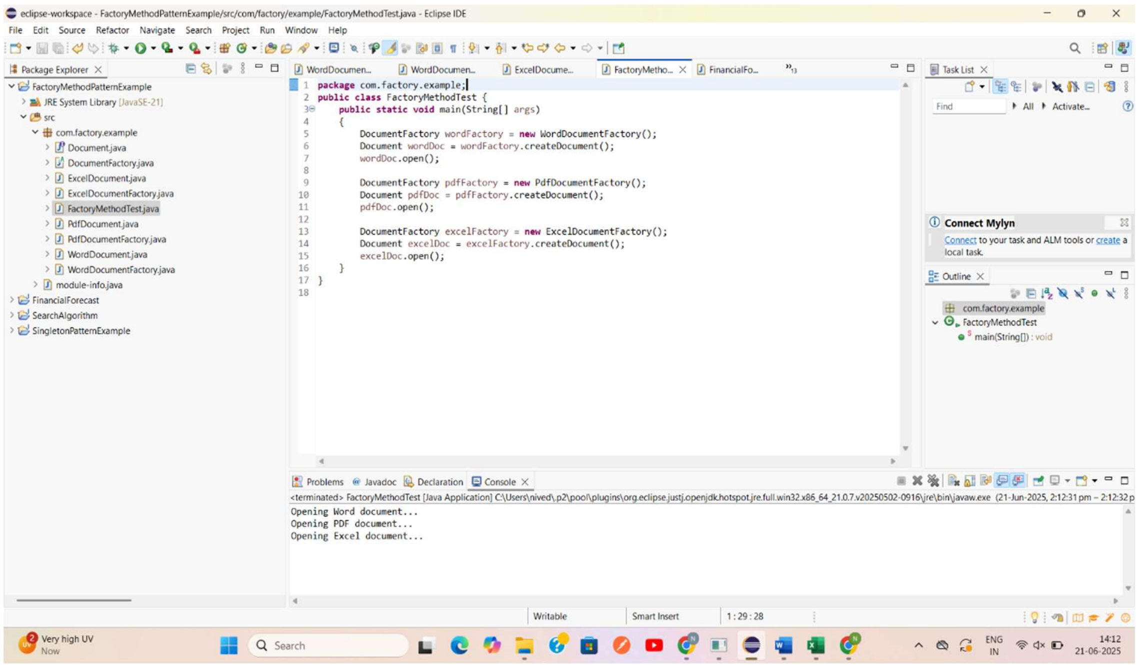


```java
package com.singleton;
public class LoggerTest
{
    public static void main(String[] args)
    {
        Logger logger1 = Logger.getInstance();
        logger1.log("This is the first message.");
        Logger logger2 = Logger.getInstance();
        logger2.log("This is the second message.");

        if (logger1 == logger2)
        {
            System.out.println("Only one instance of Logger is used.");
        } else
        {
            System.out.println("Different instances of Logger exist!");
        }
    }
}
```

```
<terminated> LoggerTest [Java Application] C:\Users\nived\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_21.0.7.v20250502-0916\jre\bin\javaw.exe (21-Jun-2025, 1:38:54 pm – 1:38:57 pm elaps
Logger Initialized
Log message: This is the first message.
Log message: This is the second message.
Only one instance of Logger is used.
```

Exercise 2:Implementing the Factory Method pattern

Output:



eclipse-workspace - FactoryMethodPatternExample/src/com/factory/example/FactoryMethodTest.java - Eclipse IDE

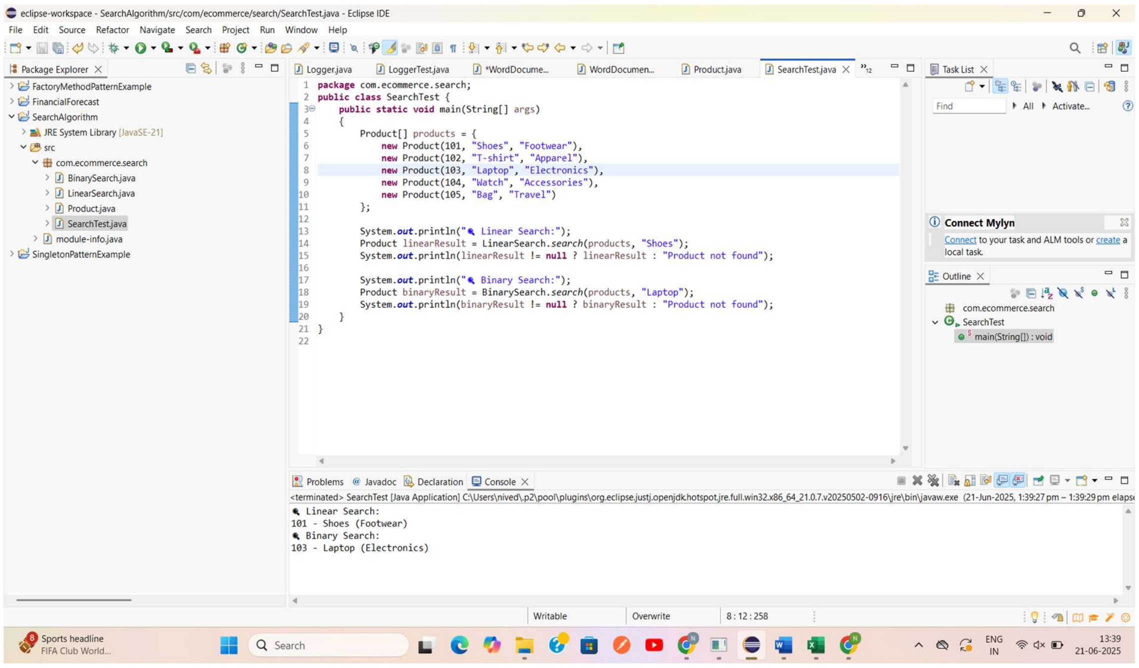
```java
package com.factory.example;
public class FactoryMethodTest {
    public static void main(String[] args)
    {
        DocumentFactory wordFactory = new WordDocumentFactory();
        Document wordDoc = wordFactory.createDocument();
        wordDoc.open();

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

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

```
<terminated> FactoryMethodTest [Java Application] C:\Users\nived\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_21.0.7.v20250502-0916\jre\bin\javaw.exe (21-Jun-2025, 2:12:31 pm – 2:12:32 p
Opening Word document...
Opening PDF document...
Opening Excel document...
```

TOPIC 2:DATA STRUCTURE AND ALGORITHMS

Exercise 2:E-commerce platform and search function

Output:



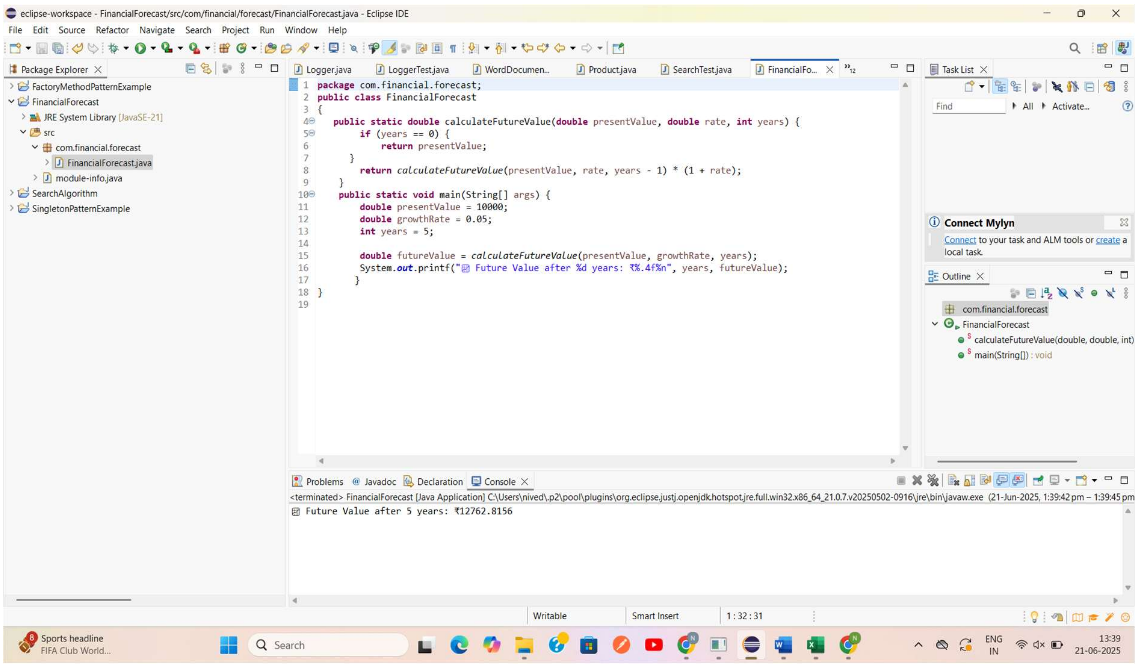
eclipse-workspace - SearchAlgorithm/src/com/ecommerce/search/SearchTest.java - Eclipse IDE

```java
package com.ecommerce.search;
public class SearchTest {
    public static void main(String[] args)
    {
        Product[] products = {
            new Product(101, "Shoes", "Footwear"),
            new Product(102, "T-shirt", "Apparel"),
            new Product(103, "Laptop", "Electronics"),
            new Product(104, "Watch", "Accessories"),
            new Product(105, "Bag", "Travel")
        };

        System.out.println("🔍 Linear Search:");
        Product linearResult = LinearSearch.search(products, "Shoes");
        System.out.println(linearResult != null ? linearResult : "Product not found");

        System.out.println("🔍 Binary Search:");
        Product binaryResult = BinarySearch.search(products, "Laptop");
        System.out.println(binaryResult != null ? binaryResult : "Product not found");
    }
}
```

```
<terminated> SearchTest [Java Application] C:\Users\nived\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_21.0.7.v20250502-0916\jre\bin\javaw.exe (21-Jun-2025, 1:39:27 pm – 1:39:29 pm elaps
🔍 Linear Search:
101 - Shoes (Footwear)
🔍 Binary Search:
103 - Laptop (Electronics)
```

Exercise 7:Financial Forecasting

Output:



eclipse-workspace - FinancialForecast/src/com/financial/forecast/FinancialForecast.java - Eclipse IDE

```java
package com.financial.forecast;
public class FinancialForecast
{
   public static double calculateFutureValue(double presentValue, double rate, int years) {
        if (years == 0) {
            return presentValue;
      }
        return calculateFutureValue(presentValue, rate, years - 1) * (1 + rate);
    }
    public static void main(String[] args) {
        double presentValue = 10000;
        double growthRate = 0.05;
        int years = 5;

        double futureValue = calculateFutureValue(presentValue, growthRate, years);
        System.out.printf("📈 Future Value after %d years: ₹%.4f%n", years, futureValue);
       }
}
```

```
<terminated> FinancialForecast [Java Application] C:\Users\nived\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_21.0.7.v20250502-0916\jre\bin\javaw.exe  (21-Jun-2025, 1:39:42 pm – 1:39:45 pm
📈 Future Value after 5 years: ₹12762.8156
```