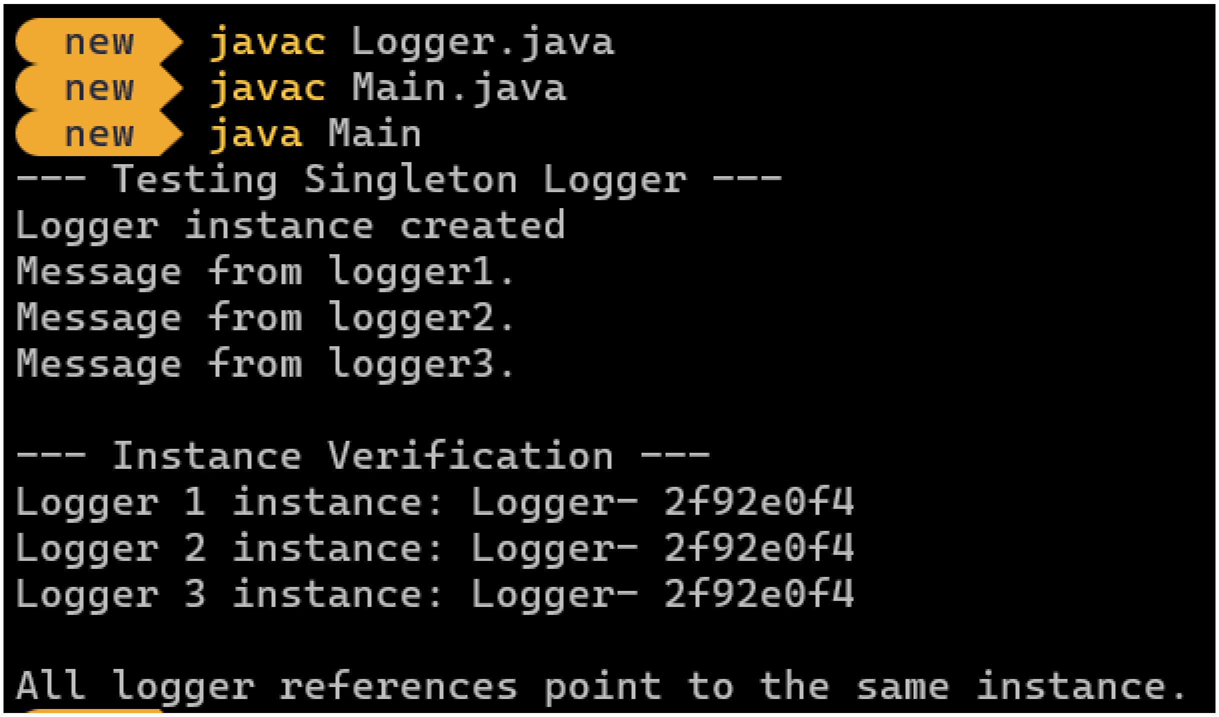
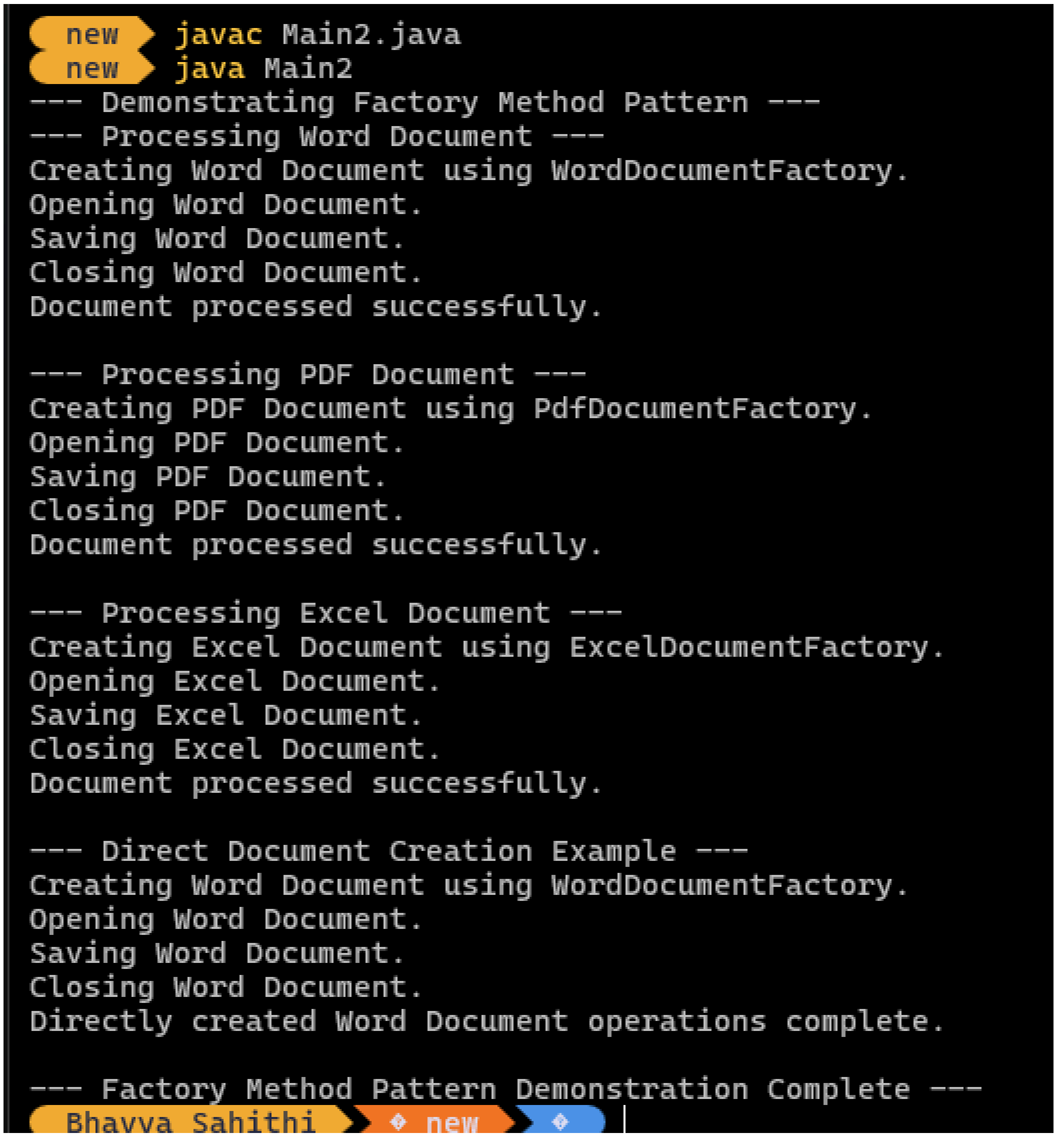
WEEK-1: OUTPUTS

Exercise 1- Implementing the Singleton pattern



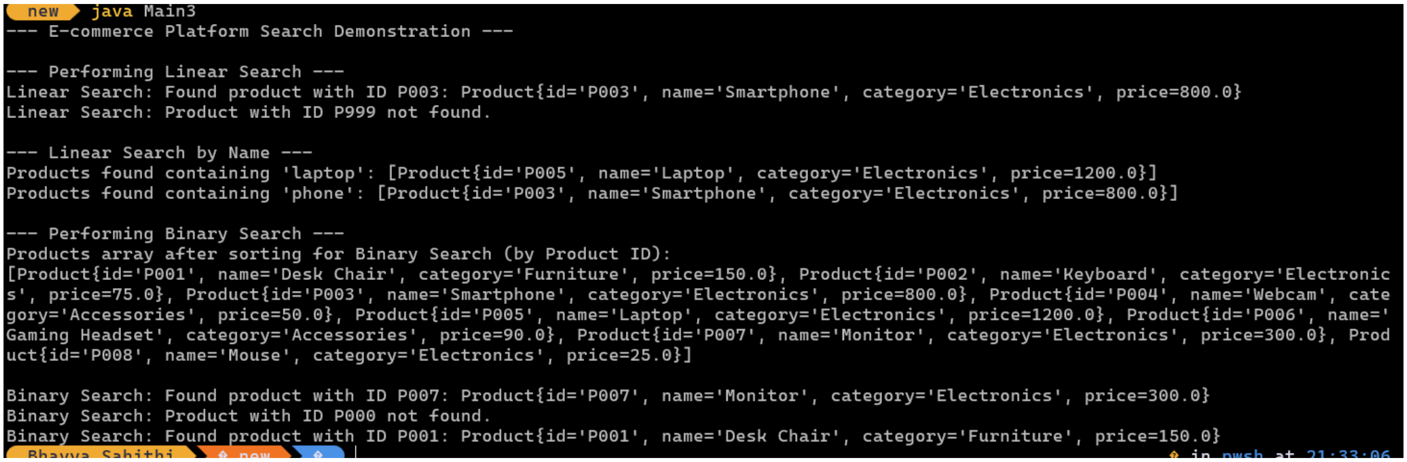
```
 new  javac Logger.java
 new  javac Main.java
 new  java Main
--- Testing Singleton Logger ---
Logger instance created
Message from logger1.
Message from logger2.
Message from logger3.

--- Instance Verification ---
Logger 1 instance: Logger- 2f92e0f4
Logger 2 instance: Logger- 2f92e0f4
Logger 3 instance: Logger- 2f92e0f4

All logger references point to the same instance.
```

Exercise 2- Implementing the Factory Method Pattern



```
 new  javac Main2.java
 new  java Main2
--- Demonstrating Factory Method Pattern ---
--- Processing Word Document ---
Creating Word Document using WordDocumentFactory.
Opening Word Document.
Saving Word Document.
Closing Word Document.
Document processed successfully.

--- Processing PDF Document ---
Creating PDF Document using PdfDocumentFactory.
Opening PDF Document.
Saving PDF Document.
Closing PDF Document.
Document processed successfully.

--- Processing Excel Document ---
Creating Excel Document using ExcelDocumentFactory.
Opening Excel Document.
Saving Excel Document.
Closing Excel Document.
Document processed successfully.

--- Direct Document Creation Example ---
Creating Word Document using WordDocumentFactory.
Opening Word Document.
Saving Word Document.
Closing Word Document.
Directly created Word Document operations complete.

--- Factory Method Pattern Demonstration Complete ---
 Bhavya Sahithi  � new  � 
```

Exercise 2: E-commerce Platform Search Function

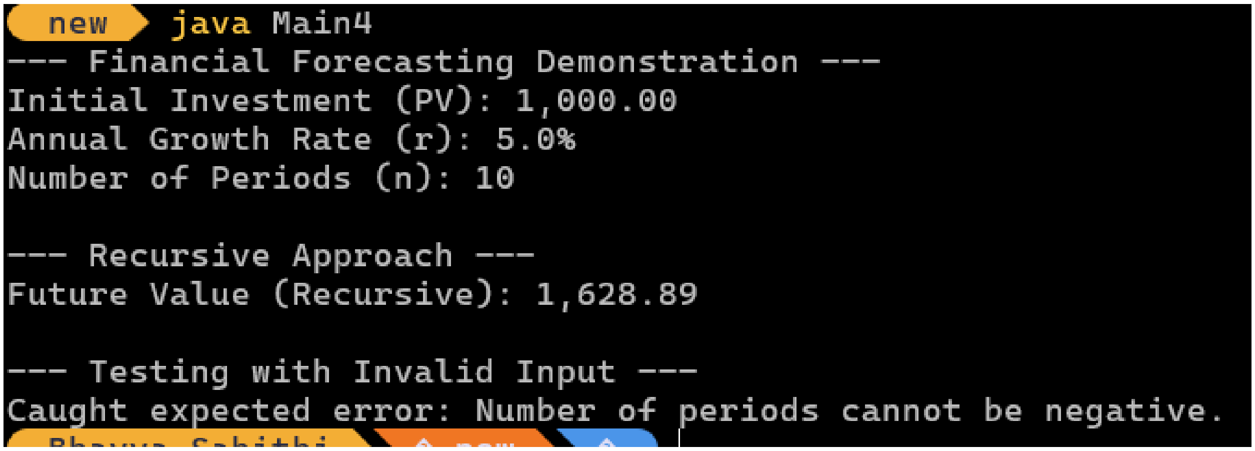


```
 new  java Main3
--- E-commerce Platform Search Demonstration ---

--- Performing Linear Search ---
Linear Search: Found product with ID P003: Product{id='P003', name='Smartphone', category='Electronics', price=800.0}
Linear Search: Product with ID P999 not found.

--- Linear Search by Name ---
Products found containing 'laptop': [Product{id='P005', name='Laptop', category='Electronics', price=1200.0}]
Products found containing 'phone': [Product{id='P003', name='Smartphone', category='Electronics', price=800.0}]

--- Performing Binary Search ---
Products array after sorting for Binary Search (by Product ID):
[Product{id='P001', name='Desk Chair', category='Furniture', price=150.0}, Product{id='P002', name='Keyboard', category='Electronic
s', price=75.0}, Product{id='P003', name='Smartphone', category='Electronics', price=800.0}, Product{id='P004', name='Webcam', cate
gory='Accessories', price=50.0}, Product{id='P005', name='Laptop', category='Electronics', price=1200.0}, Product{id='P006', name='
Gaming Headset', category='Accessories', price=90.0}, Product{id='P007', name='Monitor', category='Electronics', price=300.0}, Prod
uct{id='P008', name='Mouse', category='Electronics', price=25.0}]

Binary Search: Found product with ID P007: Product{id='P007', name='Monitor', category='Electronics', price=300.0}
Binary Search: Product with ID P000 not found.
Binary Search: Found product with ID P001: Product{id='P001', name='Desk Chair', category='Furniture', price=150.0}
 Bhavya Sahithi   new                                                                                    in pwsh at 21:33:06
```

ANALYSIS: Linear Search examines each item one by one. In the worst case, it takes O(n) time. The best case is O(1) if the item is first in order. In contrast, Binary Search is much more efficient. It repeatedly halves the search area, finding items in O(log n) time. This means even with a million products, Binary search takes less time to search compared to linear search. Binary search requires the data to be sorted first.

In the case of e-commerce platforms, Binary Search is more suitable. E-commerce sites deal with vast numbers of products, and users expect search results instantly. With hundreds of thousands of products, a binary search would give results way faster compared to a linear search.

Exercise 7: Financial Forecasting



```
 new  java Main4
--- Financial Forecasting Demonstration ---
Initial Investment (PV): 1,000.00
Annual Growth Rate (r): 5.0%
Number of Periods (n): 10

--- Recursive Approach ---
Future Value (Recursive): 1,628.89

--- Testing with Invalid Input ---
Caught expected error: Number of periods cannot be negative.
 Bhavya Sahithi   new 
```

ANALYSIS: For the recursive algorithm, the time complexity is O(n), the function recursively calls itself ‘n’ times. To optimize this we can directly use the mathematical formula to compute, which decreases the time complexity.