WEEK 1 HANDS-ON

1. Singleton Pattern

```java
public class Singleton {
    // Private static variable to hold the single instance
    private static Singleton instance;  3 usages

    // Private constructor to prevent instantiation from outside
    private Singleton() {  1 usage
        //  Print a message for testing
        System.out.println("Singleton instance created.");
    }

    // Public static method to get the single instance
    public static Singleton getInstance() {  2 usages
        if (instance == null) {
            instance = new Singleton();
        }
        return instance;
    }

    //  Main method
    public static void main(String[] args) {
        Singleton instance1 = Singleton.getInstance();
        Singleton instance2 = Singleton.getInstance();
        if (instance1 == instance2) {
            System.out.println("Both instances are the same.");
        }
    }
}
```

2. Factory method Pattern

```java
interface Shape {  no usages
    void draw();  1 usage
}
class Rectangle implements Shape {  no usages
    @Override  1 usage
    public void draw() {
        System.out.println("Drawing a Rectangle");
    }
}
class Circle implements Shape {  no usages
    @Override  1 usage
    public void draw() {
        System.out.println("Drawing a Circle");
    }
}
// Abstract creator class with the factory method
abstract class ShapeFactory {  no usages
    abstract Shape createShape();  no usages

    void drawShape() {  no usages
        Shape s = createShape();
        s.draw();
    }
}
// Concrete creators
class RectangleFactory extends ShapeFactory {  no usages
    @Override  no usages
```

```java
class RectangleFactory extends ShapeFactory {  no usages
    @Override  no usages
    Shape createShape() {
        return new Rectangle();
    }
}
class CircleFactory extends ShapeFactory {  no usages
    @Override  no usages
    Shape createShape() {
        return new Circle();
    }
}
// Main class
public class FactoryMethodDemo {
    public static void main(String[] args) {
        ShapeFactory rectangleFactory = new RectangleFactory();
        rectangleFactory.drawShape();

        ShapeFactory circleFactory = new CircleFactory();
        circleFactory.drawShape();
    }
}
```

3. E-commerce platform search function

```java
import java.util.ArrayList;
import java.util.Scanner;

// Product class to hold product details
class Product {  no usages
    String name;  no usages
    double price;  no usages

    Product(String name, double price) {  no usages
        this.name = name.toLowerCase(); // for case-insensitive search
        this.price = price;
    }

    void display() {  no usages
        System.out.println("Product: " + name + ", Price: ₹" + price);
    }
}

// Main class
public class SearchEngine {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        // List to store products
        ArrayList<Product> productList = new ArrayList<>();
```

```java
public class SearchEngine {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        // List to store products
        ArrayList<Product> productList = new ArrayList<>();

        // Add sample products
        productList.add(new Product( name: "Apple iPhone",  price: 75000));
        productList.add(new Product( name: "Samsung Galaxy",  price: 60000));
        productList.add(new Product( name: "OnePlus Nord",  price: 30000));
        productList.add(new Product( name: "Apple Watch",  price: 35000));
        productList.add(new Product( name: "Sony Headphones",  price: 12000));

        // Input search keyword
        System.out.print("Enter product keyword to search: ");
        String keyword = sc.nextLine().toLowerCase();

        boolean found = false;

        // Search in the list
        System.out.println("\nSearch Results:");
        for (Product p : productList) {
            if (p.name.contains(keyword)) {
                p.display();
                found = true;
            }
        }
```

```
"C:\Program Files\Java\jdk-24\bin\java.exe"
"-javaagent:C:\Users\shalu\OneDrive\Desktop\IntelliJ IDEA Community Edition 2025.1.2\lib\idea_rt.jar=57769" -Dfile.encoding=UTF-8 -Dsun.stdout.encoding=UTF-8 -Dsun.stderr.encoding=UTF-8 -classpath C:\Users\shalu\IdeaProjects\MyProjects\out\production\MyProjects SearchEngine
Enter product keyword to search: apple

Search Results:
Product: apple iphone, Price: ₹75000.0
Product: apple watch, Price: ₹35000.0

Process finished with exit code 0
```

4. Financial Forecasting

```java
import java.util.ArrayList;
import java.util.Scanner;

public class FinancialForecasting {
    public static double calculateForecast(ArrayList<Double> revenues, int months) {
        if (months > revenues.size()) {
            System.out.println("Not enough data to forecast.");
            return -1;
        }
        double sum = 0;
        int start = revenues.size() - months;
        for (int i = start; i < revenues.size(); i++) {
            sum += revenues.get(i);
        }
        return sum / months; // simple average
    }
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        ArrayList<Double> revenueList = new ArrayList<>();

        // Sample past revenue input
        System.out.print("Enter number of past months revenue data: ");
        int n = sc.nextInt();

        System.out.println("Enter monthly revenue data:");
        for (int i = 0; i < n; i++) {
            System.out.print("Month " + (i + 1) + ": ₹");
            double revenue = sc.nextDouble();
            revenueList.add(revenue);
        }
        System.out.print("Enter how many months to base the forecast on: ");
        int months = sc.nextInt();
        double forecast = calculateForecast(revenueList, months);
        if (forecast != -1) {
            System.out.printf("Predicted revenue for next month: ₹%.2f\n", forecast);
        }
        sc.close();
    }
}
```

```
"C:\Program Files\Java\jdk-24\bin\java.exe"
 "-javaagent:C:\Users\shalu\OneDrive\Desktop\IntelliJ IDEA Community Edition 2025.1
 .2\lib\idea_rt.jar=64574" -Dfile.encoding=UTF-8 -Dsun.stdout.encoding=UTF-8 -Dsun
 .stderr.encoding=UTF-8 -classpath C:\Users\shalu\IdeaProjects\MyProjects\out\production
 \MyProjects FinancialForecasting
Enter number of past months revenue data: 5
Enter monthly revenue data:
Month 1: ₹10000
Month 2: ₹12000
Month 3: ₹14000
Month 4: ₹15000
Month 5: ₹16000
Enter how many months to base the forecast on: 3
Predicted revenue for next month: ₹15000.00

Process finished with exit code 0
```