**Exercise 7: Financial Forecasting**

### CODE: FutureValueCalculator.java

public class FutureValueCalculator {

// Recursive method to calculate future value

public static double futureValueRecursive(double presentValue, double rate, int years) {

// Base case

if (years == 0) {

return presentValue;

}

// Recursive step

return futureValueRecursive(presentValue, rate, years - 1) \* (1 + rate);

}

}

**Main.java**

public class Main {

public static void main(String[] args) {

double pv = 10000; // Present Value ₹10,000

double rate = 0.08; // Growth rate 8%

int years = 5; // Forecasting for 5 years

// Calling recursive method from FutureValueCalculator

double futureValue = FutureValueCalculator.futureValueRecursive(pv, rate, years);

System.out.printf("Future Value after %d years: ₹%.2f\n", years, futureValue);

}

}

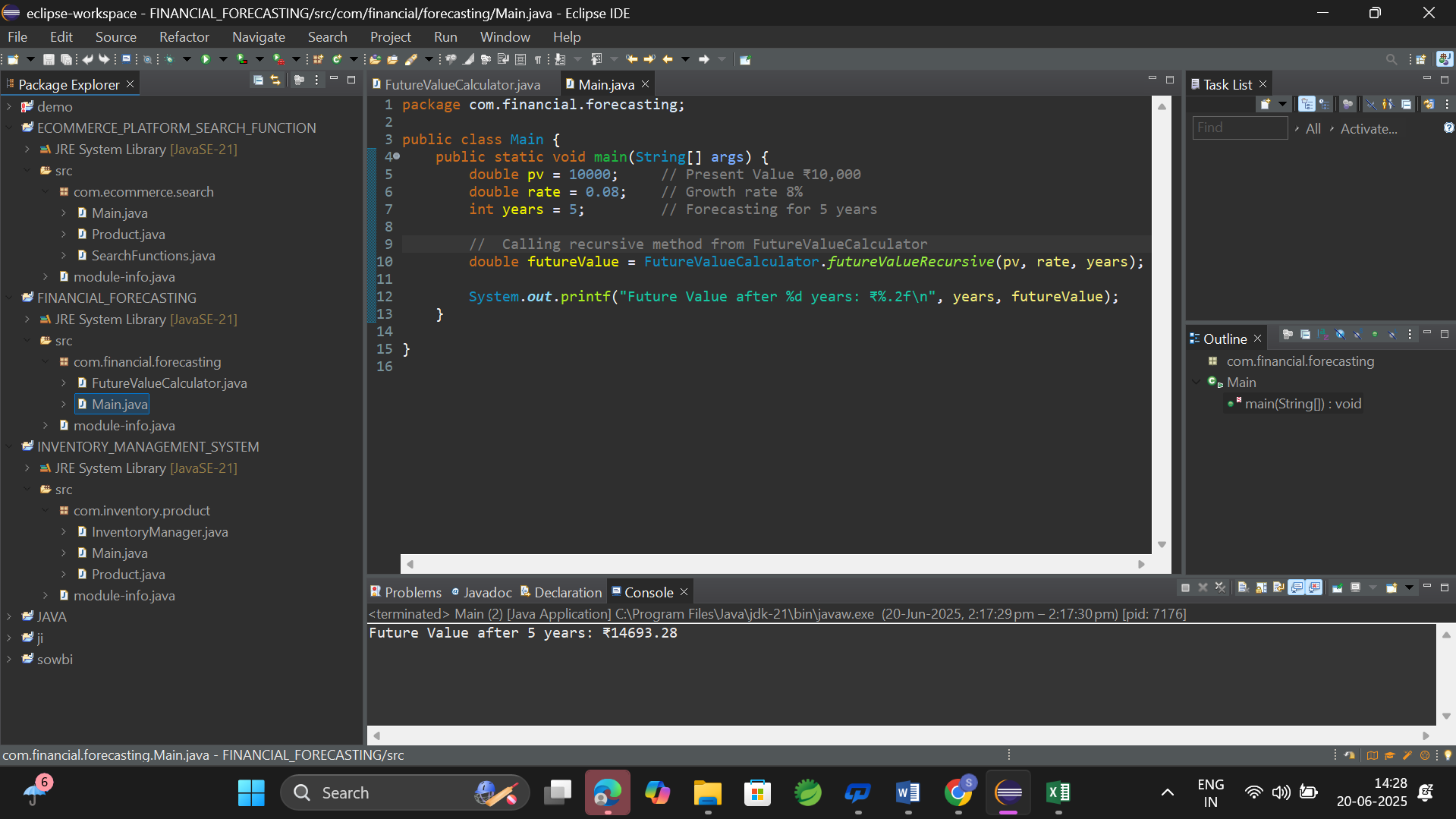
**ANALYSIS:**

| **Method** | **Time** | **Space** | **Pros** | **Cons** |
| --- | --- | --- | --- | --- |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Recursive | O(n) | O(n) | Simple to understand | Stack overhead |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Iterative | O(n) | O(1) | Efficient for large values | Slightly more code |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Formula | O(1) | O(1) | Fastest & cleanest | Might lose precision slightly |

**OUTPUT:  
  
**