**Exercise 7: Financial Forecasting**

**Scenario:**

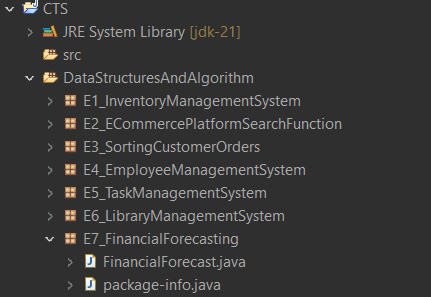
You are developing a financial forecasting tool that predicts future values based on past data.

**Steps:**

1. **Understand Recursive Algorithms:**
   * **Explain the concept of recursion and how it can simplify certain problems.**

Recursion is a method where the solution to a problem depends on solving smaller instances of the same problem. In forecasting, recursion can model compound growth year by year.

1. **Setup:**
   * **Create a method to calculate the future value using a recursive approach.**

****

1. **Implementation:**
   * **Implement a recursive algorithm to predict future values based on past growth rates.**

**FinancialForecasting.java**

package E7\_FinancialForecasting;

public class FinancialForecast {

// Recursive method to calculate future value

public static double futureValue(double presentValue, double growthRate, int years) {

if (years == 0) return presentValue;

return (1 + growthRate) \* futureValue(presentValue, growthRate, years - 1);

}

public static void main(String[] args) {

double present = 10000; // Initial investment

double growthRate = 0.08; // 8% annual growth

int years = 5; // Predict for 5 years

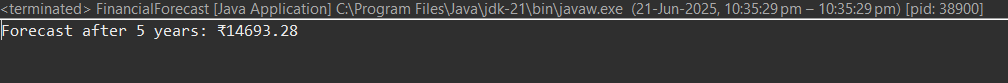
double result = futureValue(present, growthRate, years);

System.out.printf("Forecast after %d years: ₹%.2f\n", years, result);

}

}

**Output:**



1. **Analysis:**
   * **Discuss the time complexity of your recursive algorithm.**

**Time Complexity**: O(n) - One recursive call per year.

**Space Complexity:** O(n) due to recursive call stack.

* + **Explain how to optimize the recursive solution to avoid excessive computation.**

**Use Iteration**

You can replace recursion with a loop for efficiency:

