### **Exercise 7: Financial Forecasting**

**1. What is Recursion? How is it applied in the code?**  
Recursion is a technique where a function calls itself to solve smaller instances of a problem. In the code, calculateGrowth() uses recursion to compute investment growth by multiplying the current amount by (1 + rate) and calling itself with yearsLeft - 1 until the base case (yearsLeft == 0) is reached.

**2. Setup**  
The recursive method:

**public static double calculateGrowth(double currentAmount, double rate, int yearsLeft)**

* Takes currentAmount, rate, and yearsLeft.
* Returns the final amount when yearsLeft reaches 0 (base case).
* Otherwise, calculates the next year’s amount and recurses with yearsLeft - 1.

**3. Implementation**:

**double finalAmount = Forecast.calculateGrowth(5000.0, 0.05, 4);**

**Forecast.printGrowth(5000.0, 0.05, 4);**

**4. Analysis**

* **Time Complexity: O(n) (linear)** due to one recursive call per year.
* **Optimization:** Replace recursion with an iterative loop to save stack memory and improve efficiency for large inputs.While the recursion method is straightforward and easy to understand, we can actually make it even better by switching to a loop (iterative approach). This way, we can save on stack memory and improve efficiency, especially when dealing with larger inputs.

**CODE:** **Forecast.java**

**public class Forecast {**

**public static double calculateGrowth(double currentAmount, double rate, int yearsLeft) {**

**if (yearsLeft < 0) {**

**throw new IllegalArgumentException("Oops! Number of years can't be negative.");**

**}**

**if (yearsLeft == 0) {**

**return currentAmount;**

**}**

**double nextAmount = currentAmount \* (1 + rate);**

**return calculateGrowth(nextAmount, rate, yearsLeft - 1);**

**}**

**public static void printGrowth(double initial, double rate, int totalYears) {**

**System.out.println("\nHere's how your investment grows each year:");**

**for (int year = 0; year <= totalYears; year++) {**

**System.out.printf("Year %d = $%.2f%n", year, initial);**

**initial \*= (1 + rate);**

**}**

**}**

**}**

**2. Main.java**

**public class Main {**

**public static void main(String[] args) {**

**double principal = 5000.0;**

**double interestRate = 0.05;**

**int duration = 4;**

**double finalAmount = Forecast.calculateGrowth(principal, interestRate, duration);**

**Forecast.printGrowth(principal, interestRate, duration);**

**System.out.println("Starting Amount: $" + principal);**

**System.out.println("Annual Interest Rate: " + (interestRate \* 100) + "%");**

**System.out.println("Duration: " + duration + " years");**

**System.out.printf("Final Amount: $%.2f%n", finalAmount);**

**}**

**}**

**OUTPUT:- **