# **Exercise 7: Financial Forecasting**

## **Problem Statement:**

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

## **1. Understand Recursive Algorithms:**

Recursion is a programming technique where a function calls itself to solve smaller instances of a problem. It is especially useful when problems can be divided into similar subproblems.  
In financial forecasting, recursion can be used to calculate future values based on past trends or growth rates.

## **2.** **Implementation:**

* ***Recursive Forecast Code:***

public class FinancialForecast {  
  
 public static double forecast(double currentValue, double growthRate, int years) {  
 if (years == 0) {  
 return currentValue;  
 }  
 return forecast(currentValue \* (1 + growthRate), growthRate, years - 1);  
 }  
  
 public static void main(String[] args) {  
 double futureValue = forecast(1000, 0.05, 5);  
 System.out.println("Predicted future value: $" + futureValue);  
 }  
}

## **3. Analysis:**

| **Approach** | **Time Complexity** | **Space Complexity** | **Remarks** |
| --- | --- | --- | --- |
| Recursive | O(n) | O(n) | Due to function call stack |
| Iterative | O(n) | O(1) | More efficient in terms of space usage |
|  |  |  |  |

* ***Optimized iterative forecast code:***

public static double forecastIterative(double currentValue, double growthRate, int years) {  
 for (int i = 0; i < years; i++) {  
 currentValue \*= (1 + growthRate);  
 }  
 return currentValue;  
}

**OUTPUT:**

