**Exercise7: FinancialForecasting.java**

import java.util.HashMap;

public class Main{

// Recursive method to calculate future value

public static double forecastRecursive(double initialValue, double growthRate, int years) {

if (years == 0) return initialValue;

return (1 + growthRate) \* forecastRecursive(initialValue, growthRate, years - 1);

}

// Optimized recursive method using memoization

private static HashMap<Integer, Double> memo = new HashMap<>();

public static double forecastRecursiveMemo(double initialValue, double growthRate, int years) {

if (years == 0) return initialValue;

if (memo.containsKey(years)) {

return memo.get(years);

}

double result = (1 + growthRate) \* forecastRecursiveMemo(initialValue, growthRate, years - 1);

memo.put(years, result);

return result;

}

// Main method to test

public static void main(String[] args) {

double initialValue = 1000.0; // starting amount

double growthRate = 0.05; // 5% growth per year

int years = 10;

double resultRecursive = forecastRecursive(initialValue, growthRate, years);

System.out.printf("Future Value (Recursive): %.2f%n", resultRecursive);

double resultMemo = forecastRecursiveMemo(initialValue, growthRate, years);

System.out.printf("Future Value (Memoized): %.2f%n", resultMemo);

}

}

