### Financial Forecast

### What is Recursion?

**Recursion** is a programming technique where a function **calls itself** to solve a smaller part of the same problem.

### Why Use Recursion?

- **Simplifies code** for problems with repeating patterns (e.g., factorial, Fibonacci).

# Financial Forecasting Scenario

We want to forecast a **future value** (e.g., investment value, revenue) based on:

- Initial value
- Annual growth rate (as percentage)
- Number of years into the future

### 🔹 Formula:

Future Value (FV) = **P × (1 + r)^n**
Where:

- P = present value
- r = growth rate (decimal)
- n = number of years

**Implementation**

[FutureValue.cs]

```
using System;



public class Forecast

{



    public static double FutureValue(double presentValue, double rate, int
years)

    {



        if (years == 0)

            return presentValue;



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

    }

}
```

**Program.cs**

```
class Program

{

    static void Main()

    {

        double presentValue = 10000; // e.g. Rs. 10,000

        double rate = 0.05;          // 5% growth rate

        int years = 5;               // forecast for 5 years



        double futureValue = Forecast.FutureValue(presentValue, rate,
years);

        Console.WriteLine($"Future Value after {years} years:
{futureValue:C}");

    }

}
```

## Time Complexity:

`FutureValue(presentValue, rate, years)`

Each call reduces `years` by 1, so:

- Time Complexity: O(n) where `n=years`
- Space Complexity: O(n) due to the call stack (recursion depth)

---

## 🔹 How to Optimize Recursion?

🔸 Use Memoization (if overlapping subproblems exist)

Not required here since every year is a unique subproblem.

🔸 Use Iterative Approach (to reduce call stack usage)