**What is recursion?**

Recursion is a programming technique where a function calls itself to solve a problem by breaking it down into smaller, similar subproblems. It's particularly useful for problems that can be naturally expressed in terms of smaller instances of themselves, such as tree traversal or mathematical sequences like the Fibonacci sequence.

**Why** **use** **recursion**?

* Simplifiescomplexproblems**:**

Recursion can make complex problems more manageable by breaking them into smaller, easier-to-solve parts.

* Codeeleganceandreadability**:**

Recursive solutions can be more concise and easier to understand than iterative solutions for certain problems.

* Essentialforsomealgorithms**:**

Recursion is fundamental to algorithms like tree traversal, graph algorithms (depth-first search), and dynamic programming.

**How** **it** **works**:

A recursive function typically has two key components:

1.BaseCase**:**

This is the condition that stops the recursion. It's a simple case that can be solved directly without further function calls.

2**.**RecursiveCase**:**

This is the part where the function calls itself with a modified input, moving closer to the base case.

For example, calculating the factorial of a number (n!) can be defined recursively:

* Base case: if n is 0, return 1 (0! = 1)
* Recursive case: if n > 0, return n \* factorial(n-1)

**Code**:

import java.util.Scanner;

public class Main {

public static double futureValue(double principal, double rate, int years) {

if (years == 0) {

return principal;

}

return futureValue(principal, rate, years - 1) \* (1 + rate);

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.print("Financial Forecasting");

System.out.print("\nEnter the Principal: ");

double principal = sc.nextDouble();

System.out.print("Enter the Growth Rate : ");

double rate = sc.nextDouble();

System.out.print("Enter the No. of Years :");

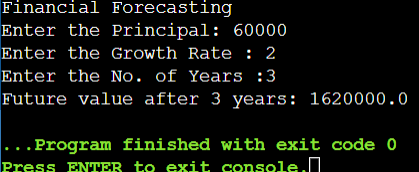
int years = sc.nextInt();

double result = futureValue(principal, rate, years);

System.out.print("Future value after "+years+" years: "+result);

}

}

** Output**: