**Exercise 7: Financial Forecasting**

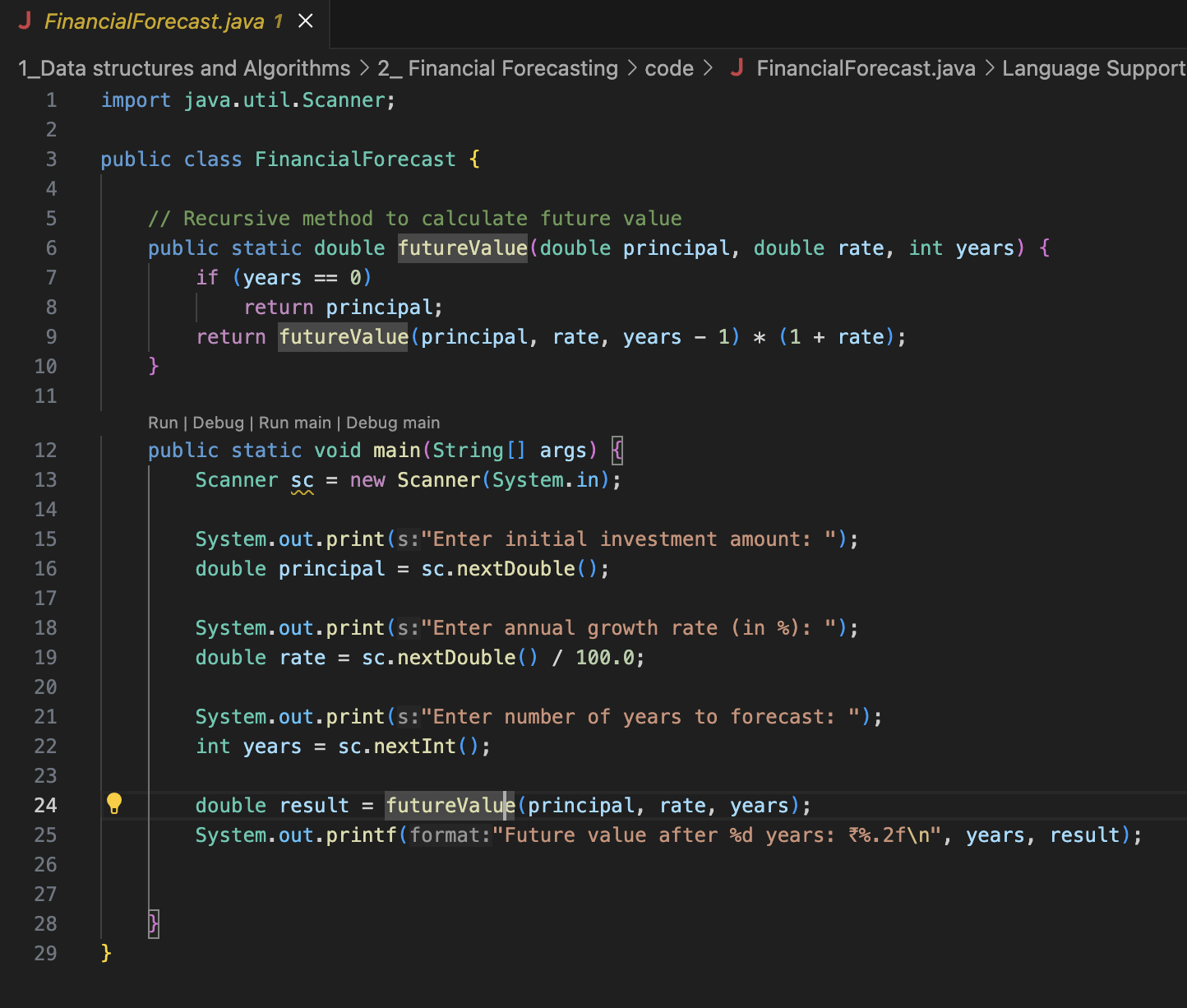
**Understanding Recursion:**

In computer programming Implementation and theory Recursion Recursion is a process in which a method (or function) makes recursive calls to solve smaller versions of the same problem until the base case halts or stops any call.

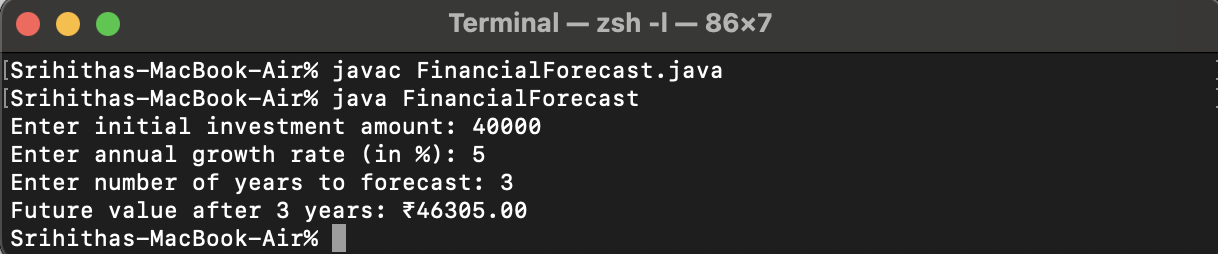
Any recursive function consists of two segments:

* Base Case -The easiest form of the problem susceptible of direct solution.
* Recursive Case -The section in which the function is called on itself tame the problem.

Recursion simplifies these problems of an algorithm:

1. It simplifies complex problems into simpler problems by splitting it into smaller and manageable sub tasks.
2. We need not write repetitive parts of a code.
3. This is useful in hierarchical and nested data (i.e, tree traversal, backtracking)
4. Recursive codes are usually shorter and easier to understand.

* The time complexity of the recursive method to calculate future value is **O(n)**.
* This function call itself once each year.
* So for example if years = n, it performs n recursive calls before reaching the base case.
* The space complexity is also **O(n)**.
* This is due to the call stack.

The output for this code would be

We can optimise this code by using a loop and eliminating recursion then we could achieve the following

* Time Complexity : O(n)
* Space Complexity : O(1) - Since there is no call stack used.