**Exercise 7: Financial Forecasting**

**1.Explain the concept of recursion and how it can simplify certain problems.**

Recursion is when a method calls **itself** to solve a problem.

It's useful when a problem can be broken into **smaller sub-problems**.

**To calculate future value for n years,:**



Instead of looping,define it recursively like:



**2.Discuss the time complexity of your recursive algorithm.**

### Time Complexity:

* Recursive function calls n times → **O(n)**

**3.Explain how to optimize the recursive solution to avoid excessive computation**

To optimize the recursive solution, use **iteration** instead of recursion to avoid deep call stacks and stack overflow. Alternatively, use **memoization** to store and reuse previously calculated results, reducing repeated computation.

**PROGRAM:**

**Forecasting.java**

package forecast;

import java.util.Scanner;

public class Forecast {

public static double FutureValue(double principal,double rate,double years){

if(years==0)

return principal;

else

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

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.print("Enter principal amount: ");

double principal = sc.nextDouble();

System.out.print("Enter annual growth rate (in %): ");

double ratePercent = sc.nextDouble();

double rate = ratePercent / 100.0; // Convert % to decimal

System.out.print("Enter number of years: ");

int years = sc.nextInt();

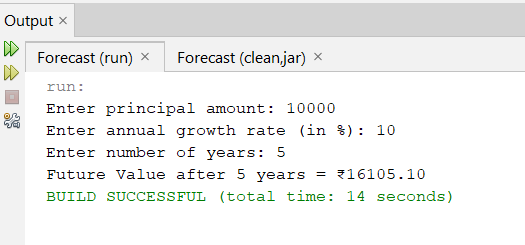
double result=FutureValue(principal,rate,years);

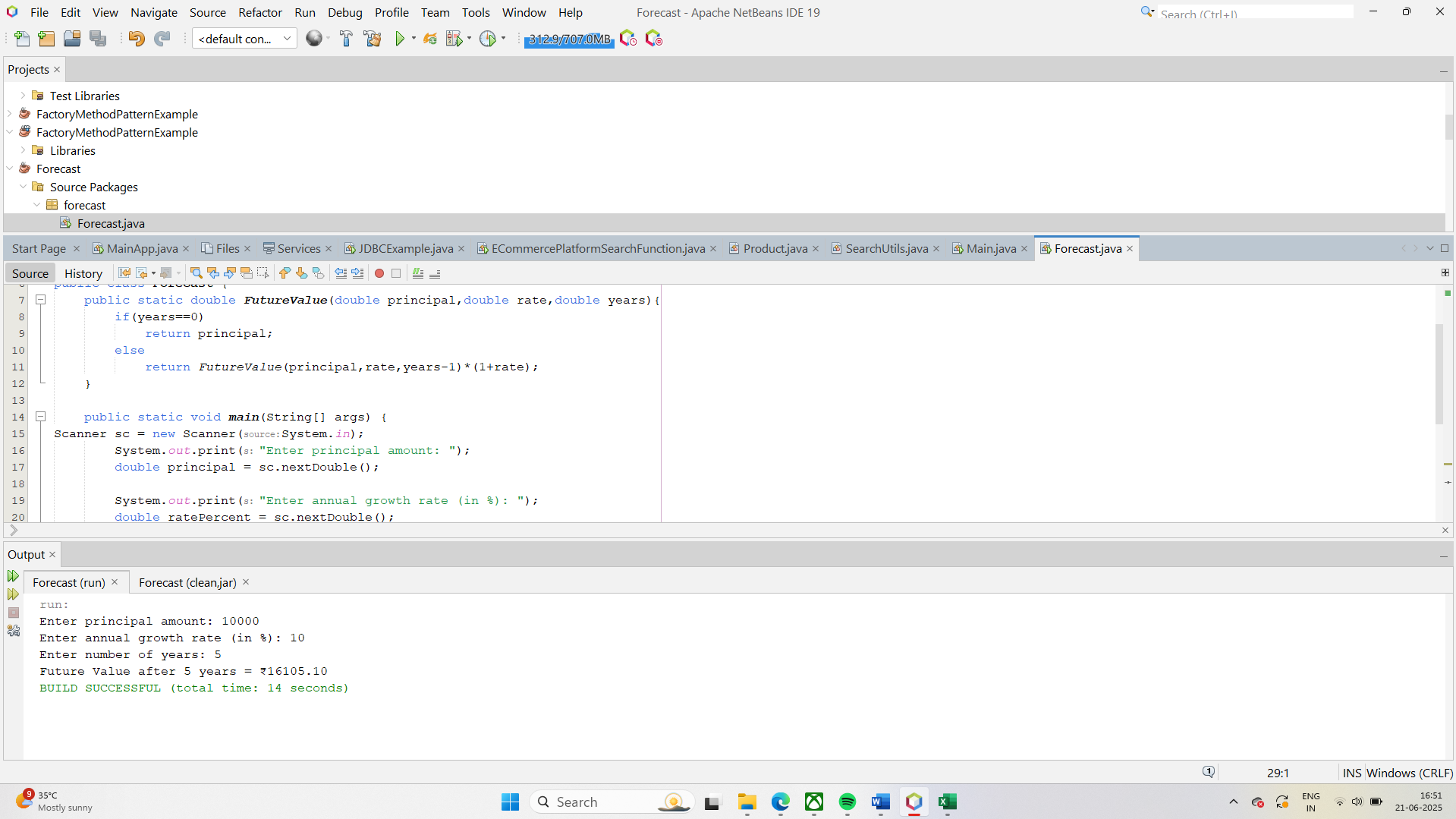
System.out.printf("Future Value after %d years = ₹%.2f\n", years, result);

}

}

**OUTPUT:**

****

****