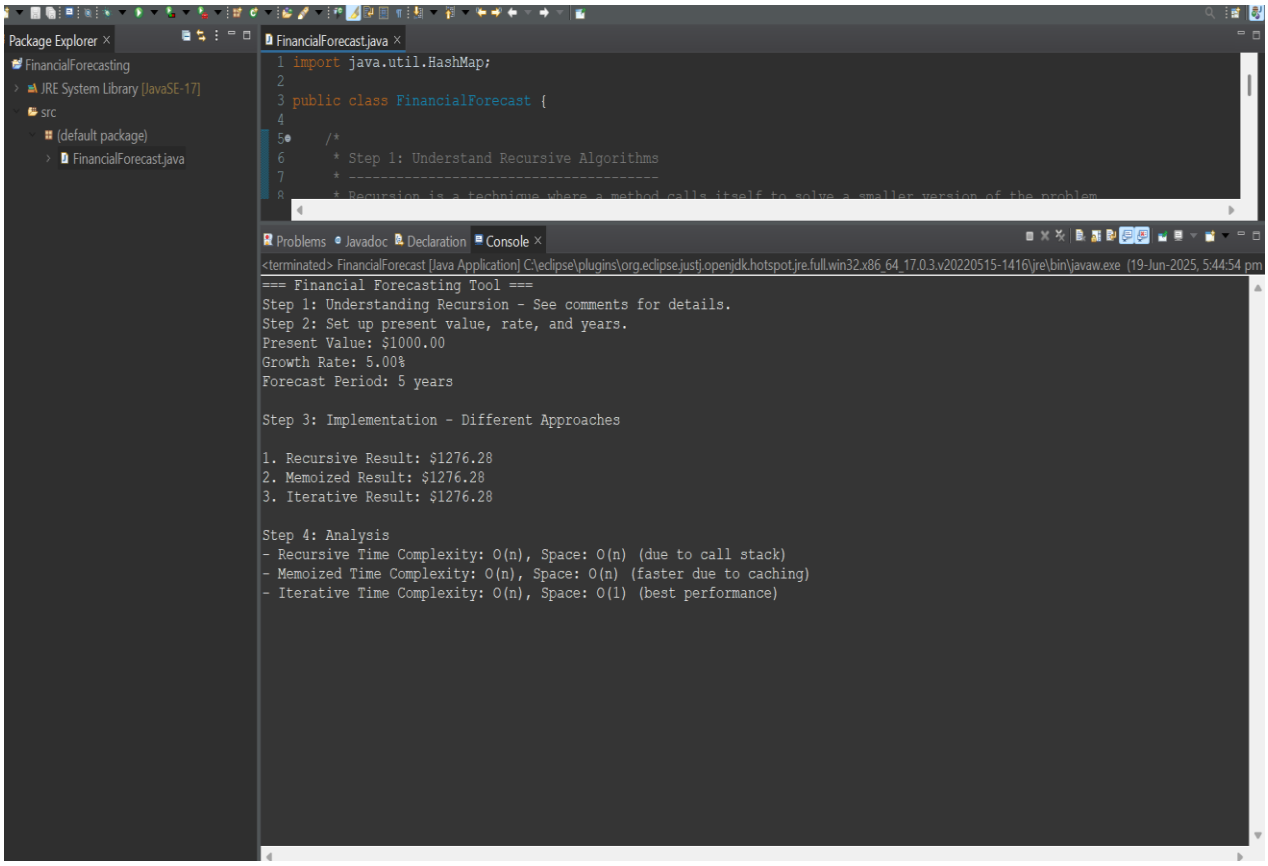
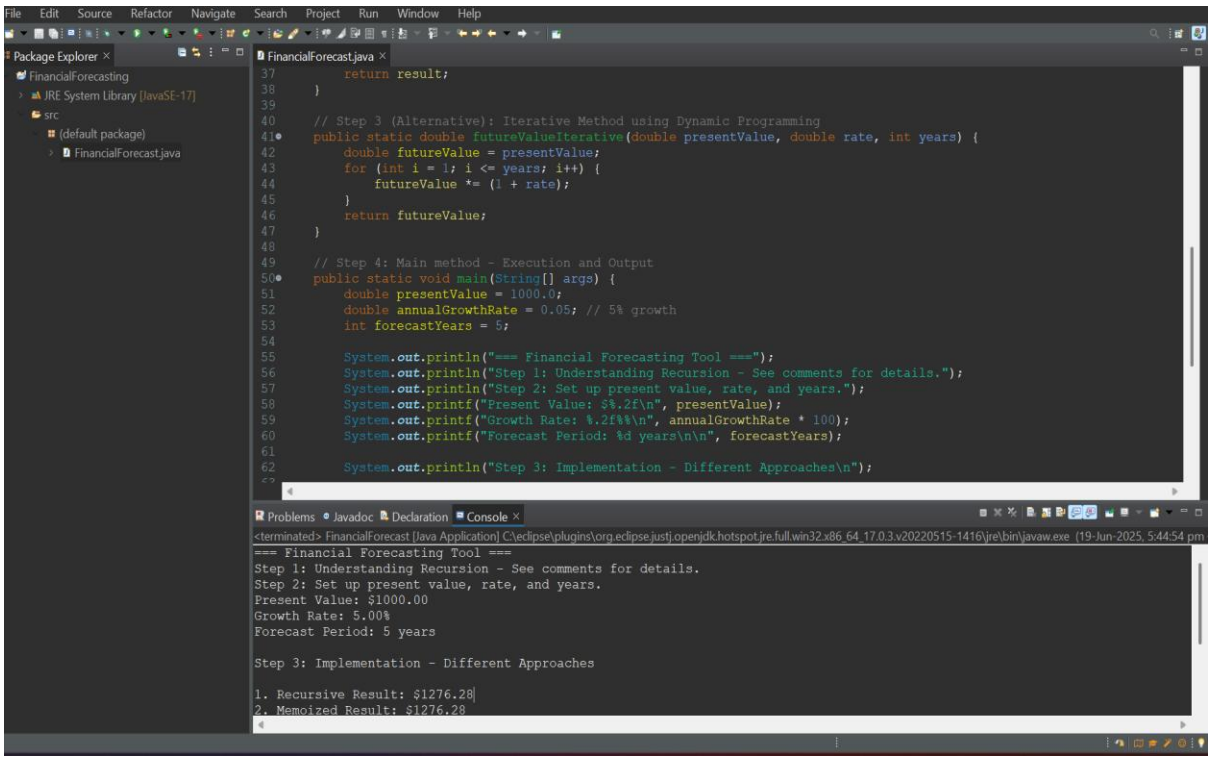
Output:-


Package Explorer
FinancialForecasting
JRE System Library [JavaSE-17]
src
(default package)
FinancialForecast.java
import java.util.HashMap;
public class FinancialForecast {
/*
* Step 1: Understand Recursive Algorithms
Problems Javadoc Declaration Console
<terminated> FinancialForecast [Java Application] C:\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.3.v20220515-1416\jre\bin\javaw.exe (19-Jun-2025, 5:44:54 pm
=== Financial Forecasting Tool ===
Step 1: Understanding Recursion - See comments for details.
Step 2: Set up present value, rate, and years.
Present Value: $1000.00
Growth Rate: 5.00%
Forecast Period: 5 years
Step 3: Implementation - Different Approaches
1. Recursive Result: $1276.28
2. Memoized Result: $1276.28
3. Iterative Result: $1276.28
Step 4: Analysis
- Recursive Time Complexity: O(n), Space: O(n) (due to call stack)
- Memoized Time Complexity: O(n), Space: O(n) (faster due to caching)
- Iterative Time Complexity: O(n), Space: O(1) (best performance)


File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
FinancialForecasting
JRE System Library [JavaSE-17]
src
(default package)
FinancialForecast.java
return result;
}
// Step 3 (Alternative): Iterative Method using Dynamic Programming
public static double futureValueIterative(double presentValue, double rate, int years) {
double futureValue = presentValue;
for (int i = 1; i <= years; i++) {
futureValue *= (1 + rate);
}
return futureValue;
}
// Step 4: Main method - Execution and Output
public static void main(String[] args) {
double presentValue = 1000.0;
double annualGrowthRate = 0.05; // 5% growth
int forecastYears = 5;
System.out.println("=== Financial Forecasting Tool ===");
System.out.println("Step 1: Understanding Recursion - See comments for details.");
System.out.println("Step 2: Set up present value, rate, and years.");
System.out.printf("Present Value: $%.2f\n", presentValue);
System.out.printf("Growth Rate: %.2f%%\n", annualGrowthRate * 100);
System.out.printf("Forecast Period: %d years\n\n", forecastYears);
System.out.println("Step 3: Implementation - Different Approaches\n");
Problems Javadoc Declaration Console
<terminated> FinancialForecast [Java Application] C:\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.3.v20220515-1416\jre\bin\javaw.exe (19-Jun-2025, 5:44:54 pm
=== Financial Forecasting Tool ===
Step 1: Understanding Recursion - See comments for details.
Step 2: Set up present value, rate, and years.
Present Value: $1000.00
Growth Rate: 5.00%
Forecast Period: 5 years
Step 3: Implementation - Different Approaches
1. Recursive Result: $1276.28
2. Memoized Result: $1276.28