

Financial Forecasting

I have used the recursive function in the class for predicting the future values.

Here is java code

1)ForeCasting.java : Code

```
package DeepSkillling;

public class Forecasting {

    public static double futureValueRecursive(double
presentValue, double rate, int years) {

        if (years == 0) return presentValue;

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

    }

    public static void main(String[] args) {

        double presentValue = 10000;

        double rate = 0.05;

        int years = 10;
```

```

        double futureValue =
futureValueRecursive(presentValue, rate, years);

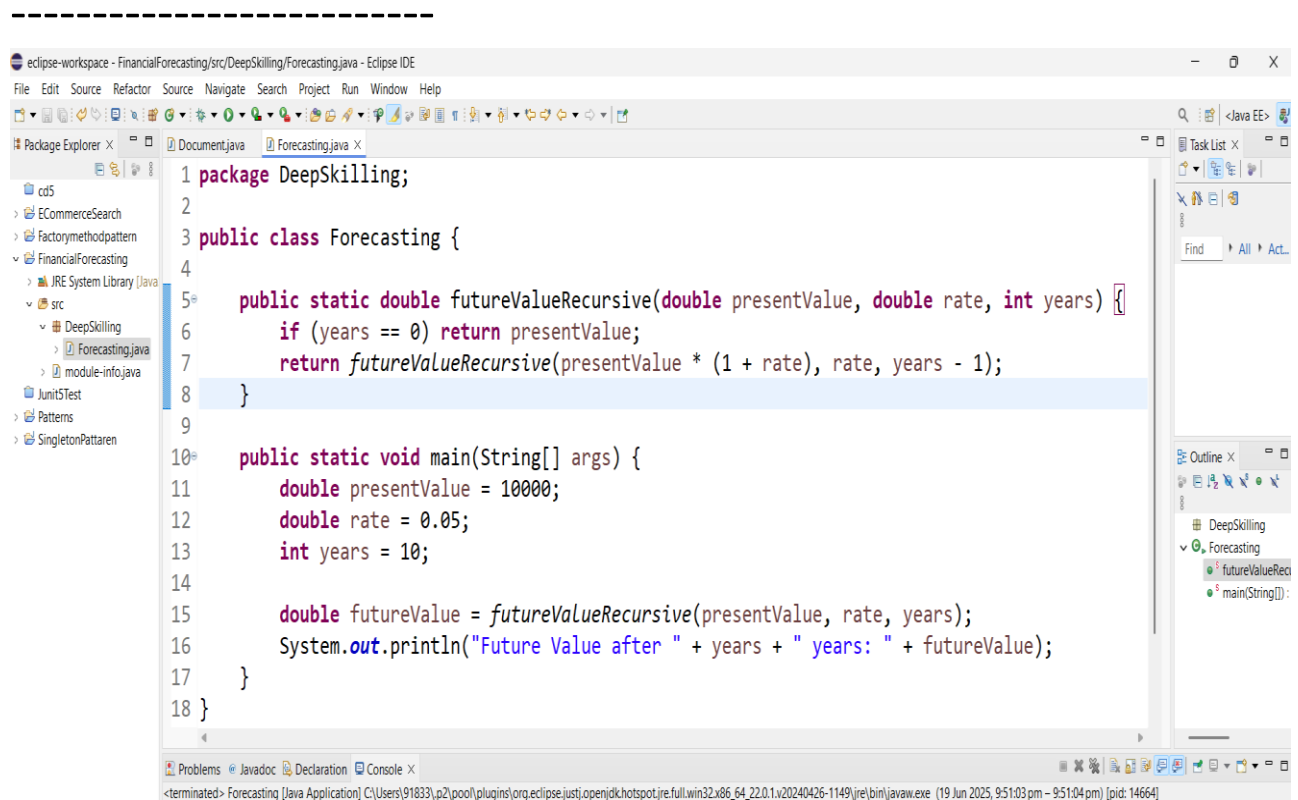
        System.out.println("Future Value after " + years + "
years: " + futureValue);

    }

}

```

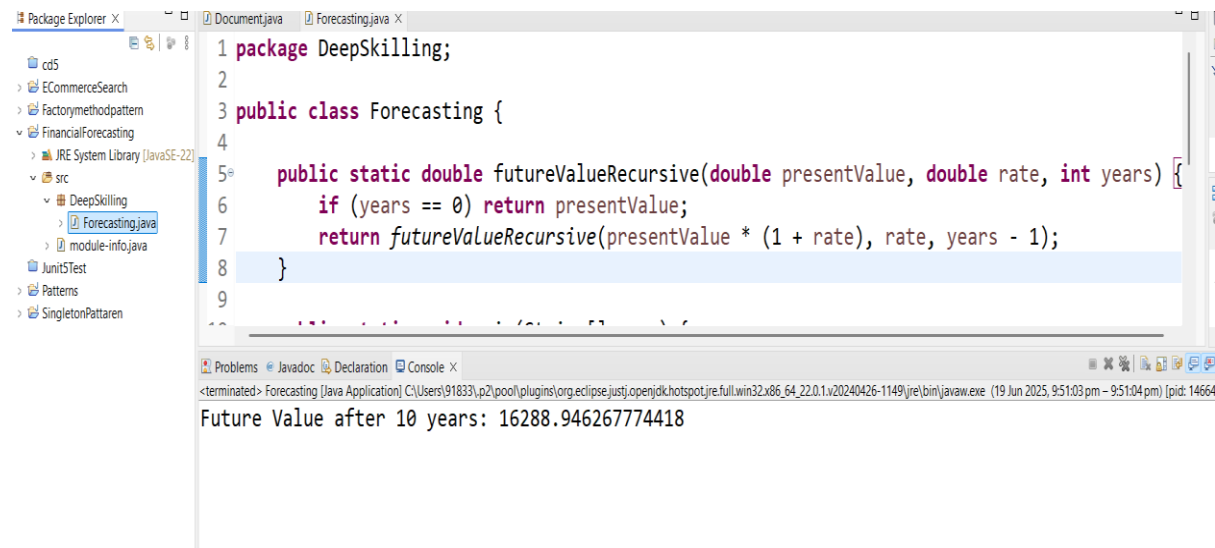
Pic From Eclipse IDE:



OUTPUT:

Future Value after 10 years: 16288.946267774418

Pic From Eclipse IDE:



```
1 package DeepSkillling;
2
3 public class Forecasting {
4
5     public static double futureValueRecursive(double presentValue, double rate, int years) {
6         if (years == 0) return presentValue;
7         return futureValueRecursive(presentValue * (1 + rate), rate, years - 1);
8     }
9
10 }
```

Future Value after 10 years: 16288.946267774418

The Time Complexity of this Recursion is $O(n)$

This will cause the Stack Overflow.

It will not suit for large Numbers.

So Recursion is not memory-efficient.

*To Reduce this we can use the For or While Loop .

*By which TimeComplexity is $O(n)$.

*But SpaceComplexity is Better $O(1)$. Unlike the Recursion Function.

*It is more Efficient for larger inputs

