Recursion Vs Iteration

Recursion

A) A recursive definition requires 2 parts
- i) A function that defines a value in terms of a lesser value .
- ii) An anchor value (i.e. an ending value) .

B) Example - define factorial

- i) n! = n*(n-1)!
- ii) 1! = 0! = 1

```
Thus  4! = 4 * 3!             defining 4! in terms of 3!
         = 4 * 3 * 2!
         = 4 * 3 * 2 * 1!
And   4! = 4 * 3 * 2 * 1      anchor 1! = 1
```

C) Implementation -

In programming : Recursion is a method which will call itself .

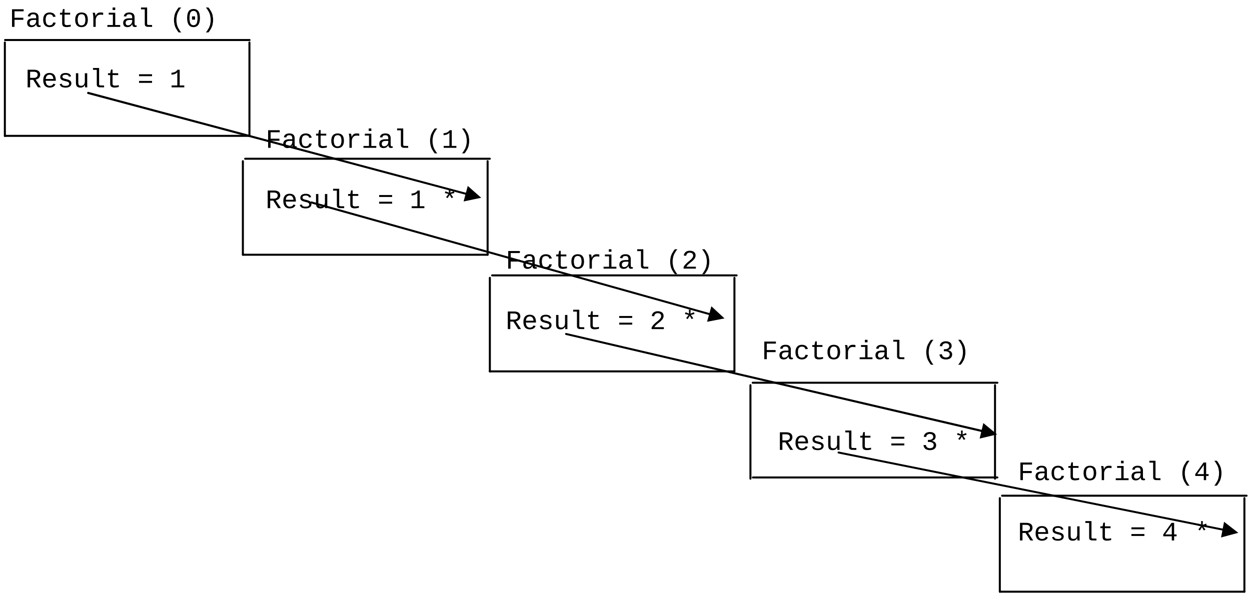
```
public int  Factorial   (int n)
   {
      int Result;

      if ( n > 0 )
         Result = n * Factorial ( n - 1 ) ;
      Else if ( n = 0 )
         Result = 1 ;
      Else
         System.out.println(" Improper value " + n + " entered" +
                              " Factorial function not executed ." ) ;
      return Result ;

   }  // End_fcn_ Factorial
```

D) Ergo

Factorial (4) calls Factorial (3), which in turn calls Factorial (2), etc .



2) Iteration

A) Iteration is the process of repeating an algorithm .
Repetition = Loop

B) Factorial iterated -

```
public int  Factorial   (int n)

   {
      int Result = 1 ;
      int J ;

      for (J = 2; J <= n; J++)
         Result = Result * J ;

      return Result ;

   }  // End_fcn_Factorial
```

3) Now Recursion Vs Iteration

A) Which to use and when ?

i) First, know there are some processes which can only be accomplished by Recursion . Especially, in Artificial Intelligence where "Back Tracking" is a necessary process .

ii) So, which is more efficient ?

a) Processor usage -

b) Memory usage -

c) Understanding logic flow -

B) Thus the rule of thumb is - Always use Iteration when possible .

4) Recursion is Fun

Find f(12) where

```
int   f     (int n)
   {
      int Result ;

      if ( n > 10 )
         Result = 1 + f(f(n - 5)) ;
      else if (n > 0)
         Result = f(n - 2) - 2 ;
      else
         Result = 1 - n ;

      return Result ;

   }  // End_fcn_f
```