I. Recursion

- A method of defining a function in terms of its own definition.
- Example:- The Fibonacci Series
    1. f(n) = f(n-1) + f(n-2)
    2. f(0) = f(1) = 1 ←——------ base case
- In programming, recursion is a method call to the same method.

II. Why write a method that calls itself?

- Recursion is a good problem-solving approach.
- Recursion algorithms are elegant, simple to understand and prove correct, easy to implement. [BUT!!! ... Recursive calls can result in an infinite loop of calls. It needs a base-case in order to stop.]

III. Recursive algorithms

- To solve a problem recursively -
    1. Break into smaller problems
    2. Solve sub-problems recursively ←——--------- Problem solving technique:
    3. Assemble sub-solutions Divide-and-Conquer

```
recursive-algorithm( input )   {
   // base-case
   if (isSmallEnough( input ))
     Compute the solution and return it
   else
      // recursive case
      Break inout into simpler instances input 1, input 2, ...
      solution1 = recursive-algorithm( input1 )
      solution2 = recursive-algorithm( input2 )
      . . . . . .
      figure out solution to this problem from solution1, solution2, . . . . . .
      return solution
}
```

IV. Example

1. Write a function that computes the sum of numbers from 1 to n.

int sum (int n)

- Use a loop
- Recursively

```
//with a loop
int sum (int n) {
    int s = 0;
    for (int i=0; i<n; i++)
        s+= i;
    return s;
}
```

```
//recursively
int sum (int n) {
    int s;
    if (n == 0) return 0;
     //else
     s = n + sum(n-1);
     return s;
}
```

HOW DOES THE RECURSION ONE WORK?