Short-term Hands-on Supplementary Course on C programming

Session 8:
Recursion

Nandakishor. V
Nitheesh Kumar. N
Sanjay Rojar U

Agenda

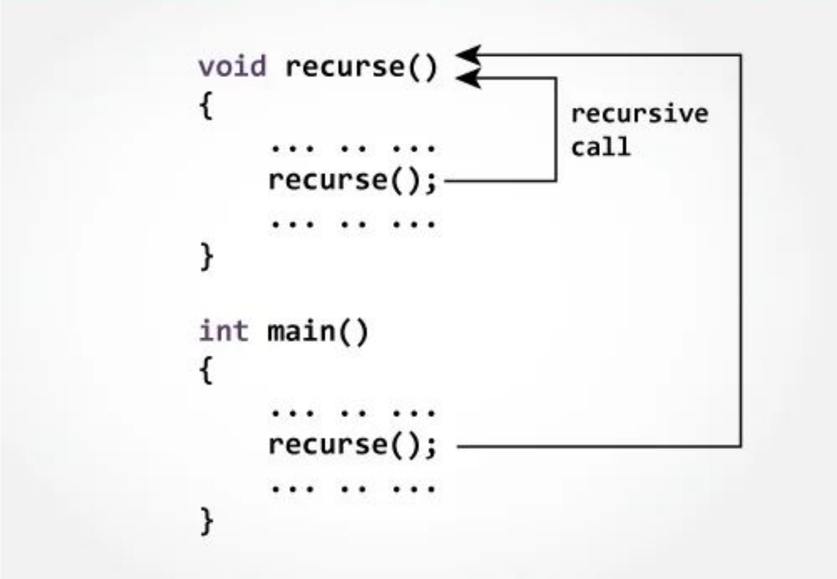
1. Recursive Functions
   a. Understanding recursion using Fibonacci Series
   b. Iteration vs Recursion

2. Variable Scopes

3. Tutorial: Implementing Recursive Functions

Recursion

Recursive Function is a function that **repeats or uses its own previous term** to calculate subsequent terms and thus forms a sequence of terms.



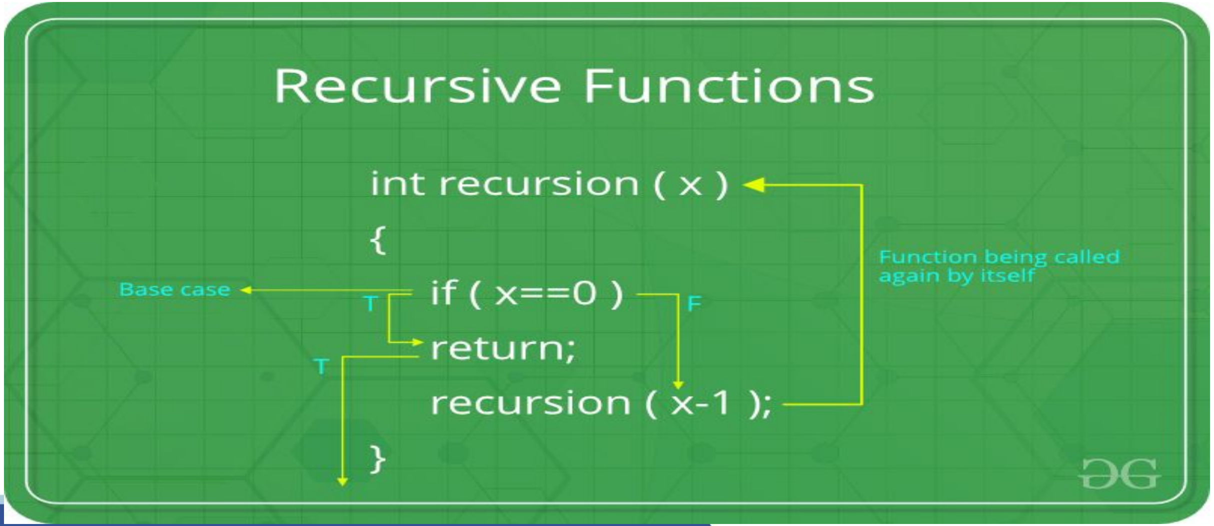
```
void recurse()
{
    ... .. ...
    recurse();
    ... .. ...
}

int main()
{
    ... .. ...
    recurse();
    ... .. ...
}
```

Structure of Recursion

Recursion involves at least **two** cases:

1.Base case:The simple case;an occurrence that can be answered directly; the case that calls that recursive calls reduce to.

2.Recursive case:a more complex occurrence of the problem that can't be directly answered, but can be described in terms of smaller occurrences of the same problem.



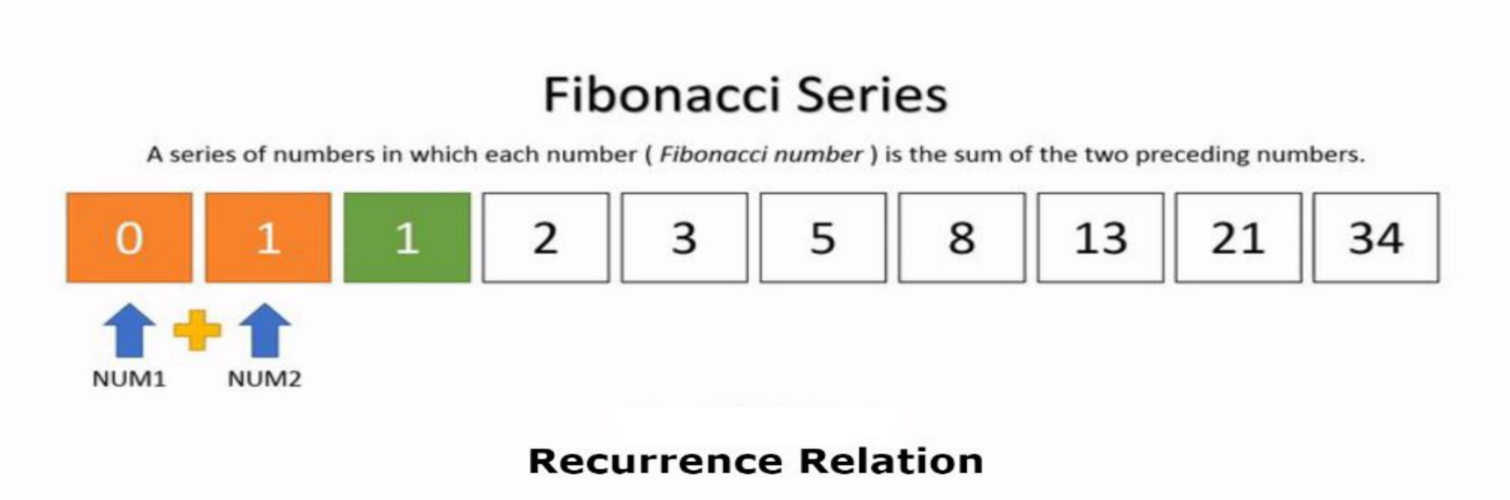
```
int recursion ( x )
{
    if ( x==0 )
    return;
    recursion ( x-1 );
}
```

Three Musts of Recursion

1. Your code must have a case for all valid inputs

2. You must have a base case that makes no recursive calls

3. When you make a recursive call it should be to a simpler instance and make forward progress towards the base case.

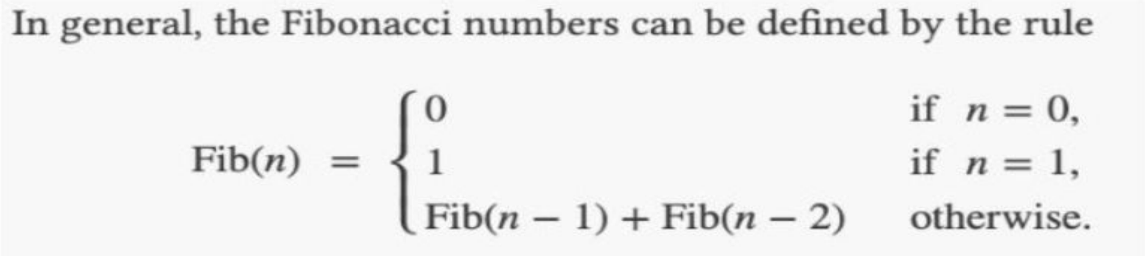


Fibonacci Series

A series of numbers in which each number (*Fibonacci number*) is the sum of the two preceding numbers.

0	1	1	2	3	5	8	13	21	34

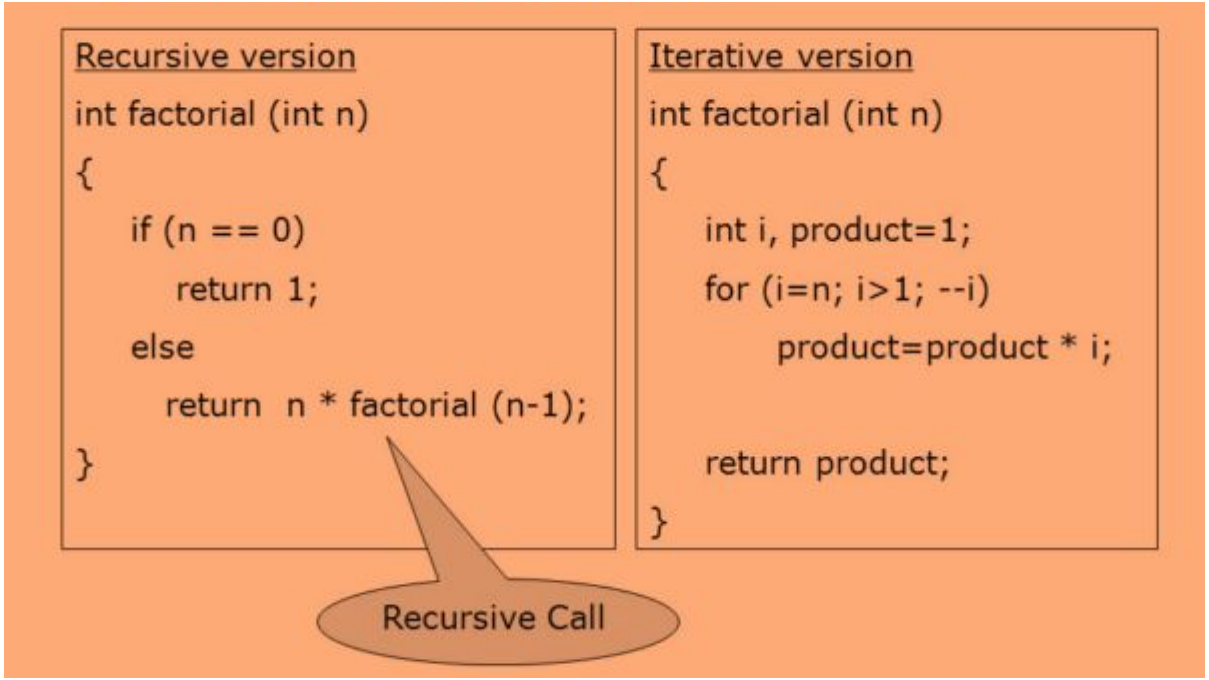
Recurrence Relation



In general, the Fibonacci numbers can be defined by the rule

$$\mathrm{Fib}(n) \;=\; \begin{cases} 0 & \text{if } \; n = 0, \\ 1 & \text{if } \; n = 1, \\ \mathrm{Fib}(n-1) + \mathrm{Fib}(n-2) & \text{otherwise.} \end{cases}$$

Iteration vs Recursion



Recursive version

```
int factorial (int n)
{
    if (n == 0)
      return 1;
    else
      return  n * factorial (n-1);
}
```

Recursive Call

Iterative version

```
int factorial (int n)
{
    int i, product=1;
    for (i=n; i>1; --i)
          product=product * i;

    return product;
}
```

Variable

Variable’s are divided on basis like:

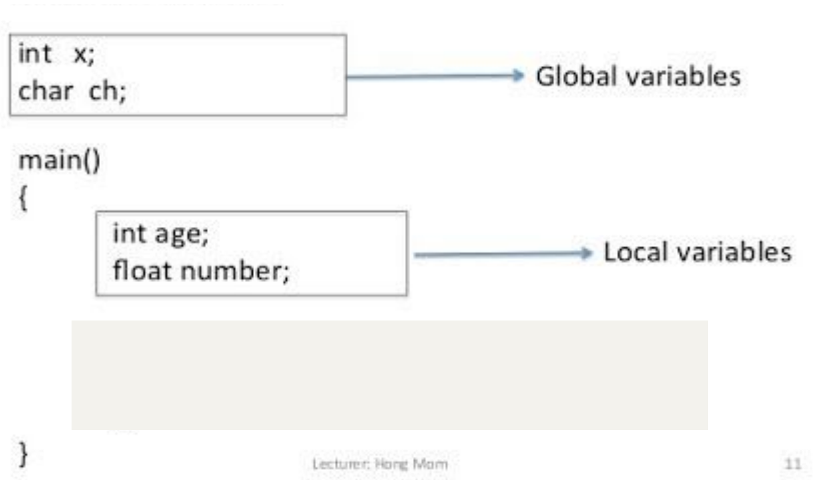
1. Scope
2. Lifetime

Scope is further divided into:

1. Local
2. Global

Lifetime can be:

1. Static
2. Automatic
3. Dynamic(Heap)



```
int  x;
char  ch;

main()
{
        int age;
        float number;



}
```

Lecturer: Hong Mom 11

Tutorial

1. Write a recursive C function to find the sum of digits of a number.

2. Write a recursive C function to implement binary search on a array.

3. Write a recursive C program to find a binary representation of a given decimal number.

4. Write a recursive C program to find the LCM and GCD of two numbers.

Any Queries !?

Thank You for attending!

Contact us regarding any questions through email

nandakishor2010608@ssn.edu.in

nitheesh2010343@ssn.edu.in

sanjayrojar2010085@ssn.edu.in