# *PROCEDURAL PROGRAMMING*

Ankit S. Chitnavis

# Introduction :-

Procedural programming is the standard approach used in traditional computer language such as **C, Pascal, FORTRAN & BASIC.**

The basic idea is to have a program specify the sequence of steps that implements a particular algorithm .

Procedural programming is a term used to denote the way in which a computer programmer writes a program.

This method of developing software, which also is called an application, revolves around keeping code as concise as possible. It also focuses on a very specific end result to be achieved.

Procedural programming creates a step by step program that guides the application through a sequence of instructions. Each instruction is executed in order.

Procedural programming focuses on processes. In procedural programming data and functions are stored in separate memory location, while in OOP data and functions are stored in same memory location.

Programs are made up of modules, which are parts of a program that can be coded and tested separately, and then assembled to form a complete program.

In procedural languages (i.e. C) these modules are procedures, where a procedure is a sequence of statements.

In C for example, procedures are a sequence of imperative statements, such as assignments, tests, loops and invocations of sub procedures. These procedures are functions, which map arguments to return statements.
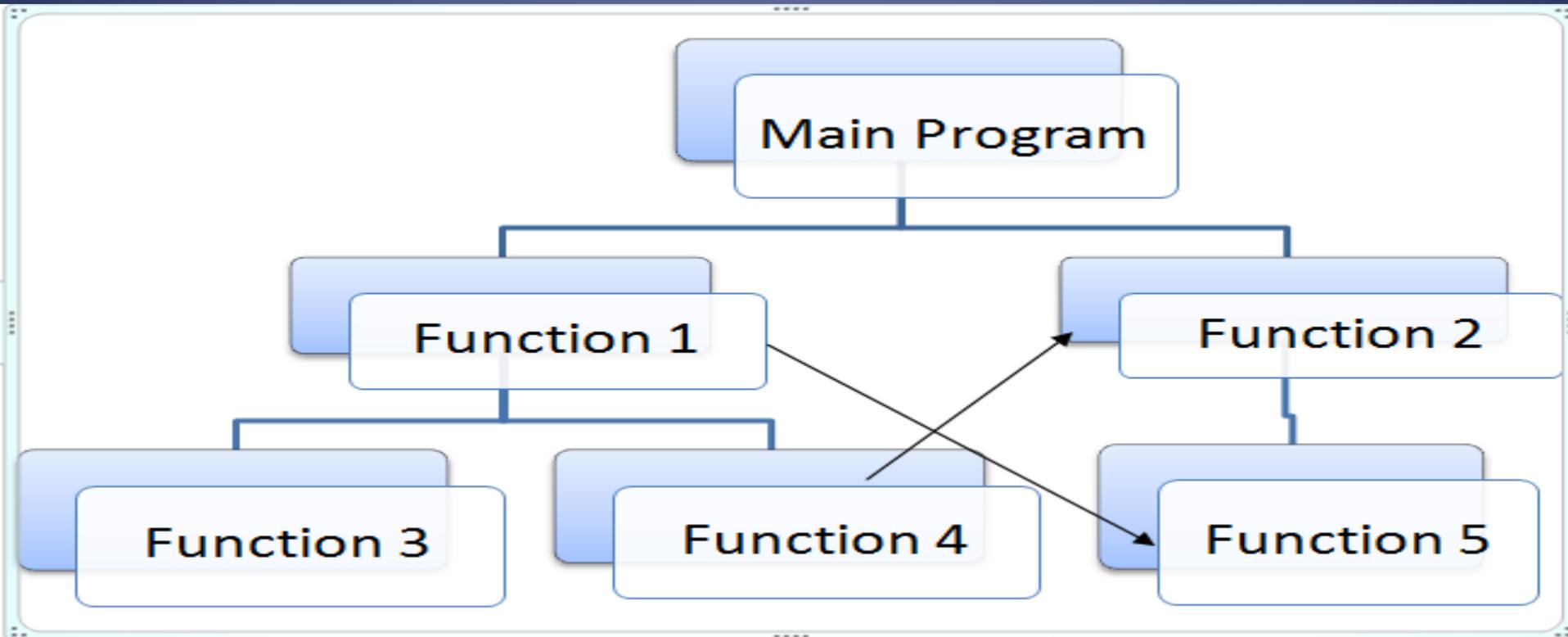
➤ The design method used in procedural programming is called **Top Down Design**.

This is where you start with a problem (procedure) and then systematically break the problem down into sub problems (sub procedures).
This is called functional decomposition, which continues until a sub problem is straightforward enough to be solved by the corresponding sub procedure.

When changes are made to the main procedure (top), those changes can cascade to the sub procedures of main, and the sub-sub procedures and so on, where the change may impact all procedures in the pyramid.

The problem with PP approach is its handling of data. PP approach gives no importance to data. By *'data'* we mean the information collected from user, the new results obtained after calculations etc.

In C, a data member must be declared GLOBAL inorder to make it accessible by 2 or more functions in the program.

*What happens when 2 or more functions work on the same data member ? If there are 10 functions in a program, all these 10 functions can access a global data member. It is possible one function may accidentally change values of this global data member.*

If this data member is a key element of the program, any such accidental manipulation will affect the whole program. It will be too difficult to debug & identify which function is causing the problem if the program is really big.

Global Data

Function 1
{Local Data}

Function 2
{Local Data}

```c
#include<stdio.h>

int Fibonacci(int);
 main()
{
    int n, i = 0, c;
    printf("Enter the number of
            terms\n");
    scanf("%d",&n);
    printf("Fibonacci series\n");
    for ( c = 1 ; c <= n ; c++ )
    {
        printf("%d\n", Fibonacci(i));
        i++;
    }
    return 0;
}

 int Fibonacci(int n)
{   if ( n == 0 )
        return 0;
else
if ( n == 1 )
    return 1;
else
 return ( Fibonacci(n-1) +Fibonacci(n-2));
}
```
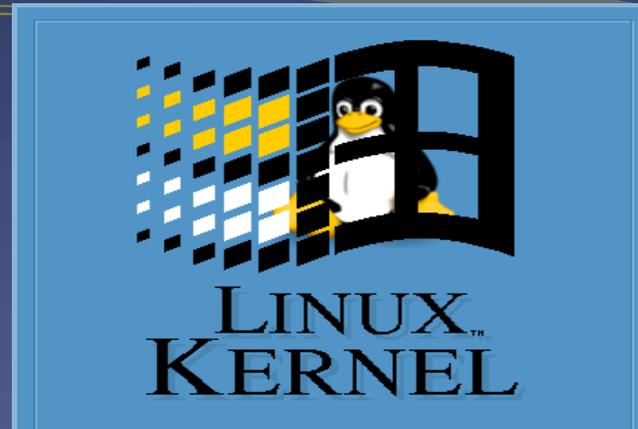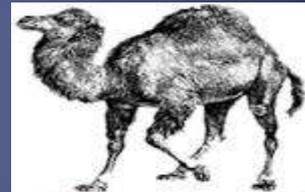
**Advantages of Procedural Programming:**

➢Its relative simplicity, and ease of implementation of compilers and interpreters.

➢The ability to re-use the same code at different places in the program without copying it.

➢An easier way to keep track of program flow.

➢ The ability to be strongly modular or structured.

➢Needs only less memory.

**Disadvantages of Procedural Programming:**

➢Data is exposed to whole program, so no security for data.

➢Difficult to relate with real world objects.

➢Difficult to create new data types reduces extensibility.

➢Importance is given to the operation on data rather than the data.