

## Functions : Definition of Functions

# What is Function in C Language?

A function in C language is a block of code that performs a specific task. It has a name and it is reusable i.e. it can be executed from as many different parts in a C Program as required. It also optionally returns a value to the calling program

So function in a C program has some properties discussed below.

- Every function has a unique name. This name is used to call function from "main()" function. A function can be called from within another function.
- A function is independent and it can perform its task without intervention from or interfering with other parts of the program.
- A function performs a specific task. A task is a distinct job that your program must perform as a part of its overall operation, such as adding two or more integer, sorting an array into numerical order, or calculating a cube root etc.
- A function returns a value to the calling program. This is optional and depends upon the task your function is going to accomplish. Suppose you want to just show few lines through function then it is not necessary to return a value. But if you are calculating area of rectangle and wanted to use result somewhere in program then you have to send back (return) value to the calling function.

C language is collection of various inbuilt functions. If you have written a program in C then it is evident that you have used C's inbuilt functions. `Printf`, `scanf`, `clrscr` etc. all are C's inbuilt functions. You cannot imagine a C program without function.

## Structure of a Function

A general form of a C function looks like this:

```
<return type> FunctionName (Argument1, Argument2, Argument3.....)
{
Statement1;
Statement2;
Statement3;
}
```

An example of function.

```
int sum (int x, int y)
{
int result;
result = x + y;
return (result);
}
```

## Advantages of using functions:

There are many advantages in using functions in a program they are:

1. It makes possible top down modular programming. In this style of programming, the high level logic of the overall problem is solved first while the details of each lower level functions is addressed later.
2. The length of the source program can be reduced by using functions at appropriate places.
3. It becomes uncomplicated to locate and separate a faulty function for further study.
4. A function may be used later by many other programs this means that a c programmer can use function written by others, instead of starting over from scratch.
5. A function can be used to keep away from rewriting the same block of codes which we are going use two or more locations in a program. This is especially useful if the code involved is long or complicated.