

C 6_1

6.1 Function Basics

6.1.1 Building Blocks of Programs

Functions like main, printf and scanf, We have already come through.

All C programs made up of one or more functions. There must be one and only one main function. All functions are at the same level - there is no nesting.

6.1.2 Return Value

Including main All functions can return a value,.

Void Return type is specified if function is returning no value. Functions can return arithmetic values (int, float etc.), pointers structures, unions, or will not return anything (void) But they cannot return an array or a function.

6.1.3 Function Parameters

Any function (as well as main) can receive some values called parameters. While calling a function we must pass values of parameters.

Format of Function :-

```
< return_type> < function_name>(parameters...)
{
}
```

Only Values of the parameters to the function at the time of calling it

If, In definition of function contains void as parameter then function will not accept any parameter.

Calling the expressions specified as arguments in a function call and the variables listed as parameters in the function definition is very usual. For example, in the following call of function, the expressions x and y*2 are the arguments passed to the function. The values of the x and y will be copied into the parameters p and q. The variable given in the function definition are called as formal argument and the expression given in the function call are called as the actual argument .

```
cal_area(a, b*2);
```