ShreeSoft
INFORMATICS

Functions in C Programming

What is Function?

A function is a block of code that performs a particular task. There are many situations where we might need to write same line of code for more than once in a program. This may lead to unnecessary repetition of code, bugs and even becomes boring for the programmer. So, C language provides an approach in which you can declare and define a group of statements once in the form of a function and it can be called and used whenever required.

Advantages of Functions

1) Avoid repetition of codes i.e. code reusability.
2) Increases program readability.
3) Divide a big problem in to small problem.
4) Reduces chances of error.
5) Debugging is easier.
6) Modifying a program becomes easier by using function.
7) Reduces the size of the code, duplicate set of statements are replaced by function calls.

Types of Function

1) In-built Function or Library Functions
2) User Defined Functions

In-built Function or Library Functions

The functions are provided by compiler through header files is called as inbuilt or library functions. If we want to use in built functions in program then we just need to include header files of that inbuilt functions in program.

Example
stdio.h→printf(), scanf(), getch(), putch() etc.
math.h→ pow(), sqrt(), sin(), cos(), log() etc.
ctype.h → isalpha(), isdigit(), toupper(), tolower() etc.

User Defined Functions

The functions that we create in a program are known as user defined functions or in other words you can say that a function created by user is known as user defined function. If we want to create user defined then we need to do following 3 things in program.

1) Function Declaration
2) Function Definition
3) Function Calling

Function Declaration: The way we can declare variable in program before it can be used similarly we can also need to declare user defined function in program. The function declaration is also called as function prototype.

Syntax:

    returntype functionname ([parameter list]);

Where

returntype – It can be specified the type of value which we want return by functions

functionname – It can be specified the name of function which is declare according to rule of idenfier.

Parameter list – This will be optional. Here we can specified input type and no. of input which we can passed to function definition.

Note: A function declaration doesn't require name of parameter to be provided, only type of the parameters can be specified.

Function Definition:

Syntax:

```
returntype functionname ( [parameter list] )
{                             └──────────→ Formal parameter
        --------  ⎫
        --------  ⎬  Function Body
        --------  ⎭
}
```

Function Calling:

Syntax:

```
functionname ([parameters])
                 └───────→ Actual Parameter
```

Scope and Life of Variable:

1) **Local Variable:** The variable declare within any function is called as Local variable. It can be accessible only within function in which they can be declared. It can be created on memory when function execution starts and vanish from memory when execution will be finished.
2) **Global Variable:** The variable declare all above the function and after #include pre-processor is called as Global variable. It can be accessible throughout the program i.e. any function in the program. It will be created on memory when program execution starts and vanish from memory when program execution will be finished.
3) **Formal Parameter:** The variable declare within any function header is called as Formal parameter. Its scope and life is similar to local variable.
4) **Actual Parameter:** The variable which we can passed at the time of function calling is called as Actual Parameter.