**Functions**

* A function is a set of statements that take inputs, do some specific computation and produces output.
* Helps reduce code redundancy.
* the parameters passed to function are called ***actual parameters***.
* The parameters received by function are called ***formal parameters***
* ***Pass by Value:*** In this parameter passing method, values of actual parameters are copied to function’s formal parameters and the two types of parameters are stored in different memory locations. So any changes made inside functions are not reflected in actual parameters of caller.
* ***Pass by Reference*** Both actual and formal parameters refer to same locations, so any changes made inside the function are actually reflected in actual parameters of caller.
* Exit(0) is used to exit the program and only constructor and if return 0 is used then destructor is also called.
* C does not support function overloading.
* We can return multiple values from a function by using

1. Pointers
2. Structures
3. Arrays

* The mystery behind exit() is that it takes only integer args in the range 0 – 255 . Out of range exit values can result in unexpected exit codes. An exit value greater than 255 returns an exit code modulo 256.
* If there is no return type of function the default is int.
* If a function is called before its declaration then the default return type is int.
* In C, [exit()](https://www.geeksforgeeks.org/understanding-exit-abort-and-assert/) terminates the calling process without executing the rest code which is after the exit() function.
* \_Exit() – is used to terminate the program without any cleanup.
* printf("%s",\_\_func\_\_); - this statement will return the function name in which it is executed.
* \_\_FILE\_\_ - it will give the address of the file whre the file is saved.
* \_\_LINE\_\_ - it will give the line number where the line is written.