**Important Points**

* Modulus operator can’t be applied to floating numbers.
* In C language an arithmetic operation between two int variables always results in a integer type value, therefore 2/5 results in zero.
* 5%2 = 1; -5%2 = -1; 5%-2=1
* In C language, any non-zero value is considered to be true.
* char values are allowed in switch as they can be easily evaluated to an integer.
* Program execution starts from the main function in the sequence the instructions the written.
* Parameters are value placeholders in the function definition whereas arguments are actual values passed to a function to make a call.
* A function can return only one value at a time.
* Here typecasting to float is important to get the return in float type.

float average(int a, int b, int c){

float result;

result = (float)(a + b + c)/3;

return result;

* An array can contain only of similar data type.
* Structures can be used to return more than one value from a function by passing structures to a function.
* When a structure is instantiated, its elements are stored in continuous memory locations.
* malloc function allocates the memory in heap and returns a pointer of void type. In case if it is unable to allocate memory due to insufficient memory, it returns NULL pointer.
* Low level means nearest to the hardware, and a high level means farther from the hardware with a lot of layers of abstraction.
* # A string can either be created by **char \*ptr** or by **char ptr[]**. But an array cannot be created by int \*ptr = {2,4,5}. But the variable name of both – string and array works as a address of first element of string and array respectively.
* If a pointer is created pointing to array or char array then the pointer can be used to access the elements of array or string respectively.
* While using %c as format specifier in printf() or scanf() we need to use directly give the element of string but in case of %s we need to give the address of string from whichever index we want to print the string.

%c 🡪 value

%s 🡪 address

* In gets() and puts() functions we need to pass the address.
* While prototyping array as a parameter in a function, we can write either array or pointer pointing to array, but while passing argument to that function at the time of calling - we need to pass the address of array.