Pointer

Pointer is a variable that stores the address of another variable. They can make some things much easier, help improve your program's efficiency, and even allow you to handle unlimited amounts of data.

C Pointer is used to allocate memory dynamically i.e. at run time. The variable might be any of the data type such as int, float, char, double, short etc.

Syntax: Pointers require a bit of new syntax because when you have a pointer, you need the ability to both request the memory location it stores and the value stored at that memory location. data_type *ptr_name;

Example:
int-a; char *a;
Where is used to denote that "a" is pointer variable and not a normal variable.

Key points to remember about pointers in C:

* Normal variable stores the value whereas pointer variable stores the address of the variable.

* The content of the pointer always be a whole number i.e. address.

* Always C pointer is initialized to null, i.e. int *p = null.

* The value of null pointer is 0.

* & symbol is used to get the address of the variable.

*symbol is used to get the value of the variable that the pointer is pointing to.

* If pointer is assigned to NULL, it means it is pointing to nothing.

* Two pointers can be subtracted to know how many elements are available between these two pointers.

*But, Pointer addition, multiplication, division are not allowed.

Example program for pointer in C:

```
#include
int main()
{
int *ptr, q;
q = 50;
/* address of is assigned to ptr */
ptr = &q;
/* display q's value using ptr variable */
printf("%d", *ptr);
return 0;
}
```

NULL Pointer

It is always a good practice to assign a NULL value to a pointer variable in case you do not have exact address to be assigned. This is done at the time of variable declaration. A pointer that is assigned NULL is called a null pointer.
Eg: int *ptr = NULL;

The value of ptr is 0

Pointer Arithmetic

As you understood pointer is an address which is a numeric value; therefore, you can perform arithmetic operations on a pointer just as you can a numeric value. There are four arithmetic operators that can be used on pointers: ++,--, +, and -

Example
ptr++;
ptrptr+21;
ptr-10;

If a char pointer pointing to address 100 is incremented (ptr++) then it will point to memory address 101

Pointers vs Array

Pointers and arrays are strongly related. In fact, pointers and arrays are interchangeable in many cases. For example, a pointer that points to the beginning of an array can access that array by using either pointer arithmetic or array-style indexing.

```
int main()
{
int var[3] = {1, 2, 3);
int *ptr;
printf("%d\n", *ptr);
ptr++;
printf("%d\n", ptr);
return 0;
}
```

this code will return:

1

2

Pointers to Pointer

A pointer to a pointer is a form of multiple indirection or a chain of pointers. Normally, a pointer contains the address of a variable. When we define a pointer to a pointer, the first pointer contains the address of the second pointer, which points to the location that contains the actual value.

```
int main()
{
int var;
int *ptr;
int **pptr;
var = 3000;
ptr = &var;
pptr = &ptr;
printf("Value of var:%d\n", var);
printf("Value available at optr:
%d\n",*ptr);
printf("Value available at **pptr:
%d\n",**pptr);
return 0;
}
```

This code will return
Value of var :3000
Value available at *ptr: 3000
Value available at **pptr: 3000