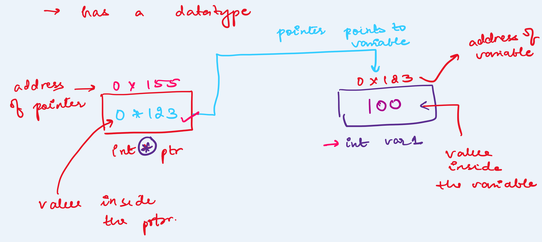
Pointers

→ A pointer refers to a variable that holds the address to another variable

→ has a datatype



& → reference pointer

(x) → variable

&x → returns the address of the variable

Syntax :

datatype * variable_name ;

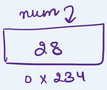
```
int *a;
chor *c;
float *C;
```

Reference pointer (&) → returns the variable address

Deference pointer (*)

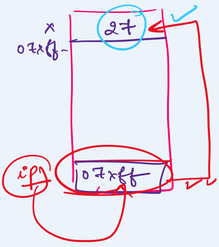
→ helps us get the value stored in memory address.

For eg :



& num → 0 x 234

* num → 5



x → 27

Pointers & Arrays

The array name itself denotes the base address of the array.

To assign the address of an array to a pointer, you should not use an ampersand (&)

p = arr ✓✓

p = &arr ; ××

```
int arr [20];
int * ip ;
ip = arr ;
ip = arr2;
```

arr2

Null pointers

If there is no exact address that is to be assigned, then the pointer variable can be assigned a NULL. It should be done during the declaration.

Its value is zero.

Pointer of variables

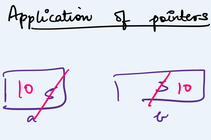
With c++ you can manipulate data directly from computer memory.

The memory space can be assigned or re-assigned as one wishes.

Pointer variables point to a specific address in the computer's memory pointed to by another variable.

```
int *p;
int* p;
```

Application of pointers



Dynamic memory allocation