

Pointers

Memory Address

int a = 10

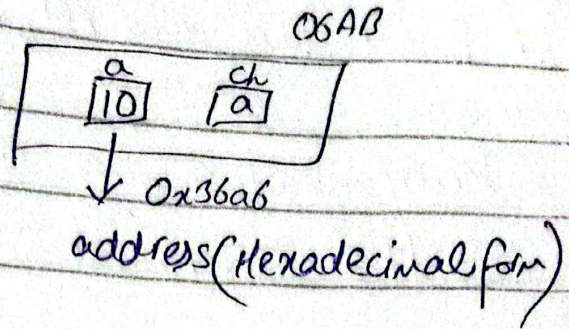
char ch = 'a'

Printing the address

Address of operator

3

3a = 0x36ab (Address of a)

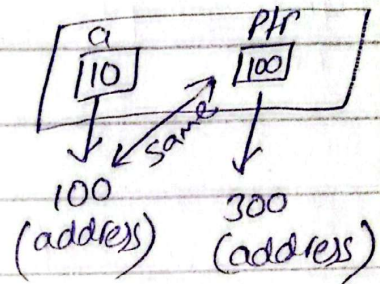


Pointers

"Special variables that stores addresses of other variable."

int a = 10

int *ptr = &a



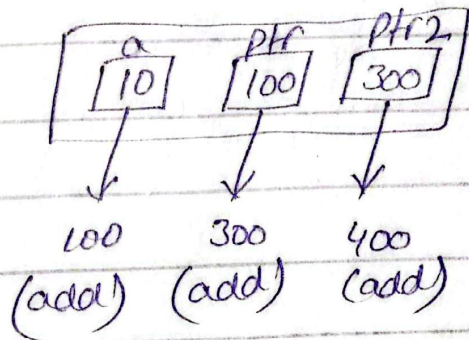
Pointer to Pointer

int a = 10

int *ptr = &a

int **ptr2 = &ptr

parent



Dereference operator

value at the address

(*)

*(&a) → 10

*(&ptr) → 10

*(&ptr2) → 100

**(&ptr2) → 10

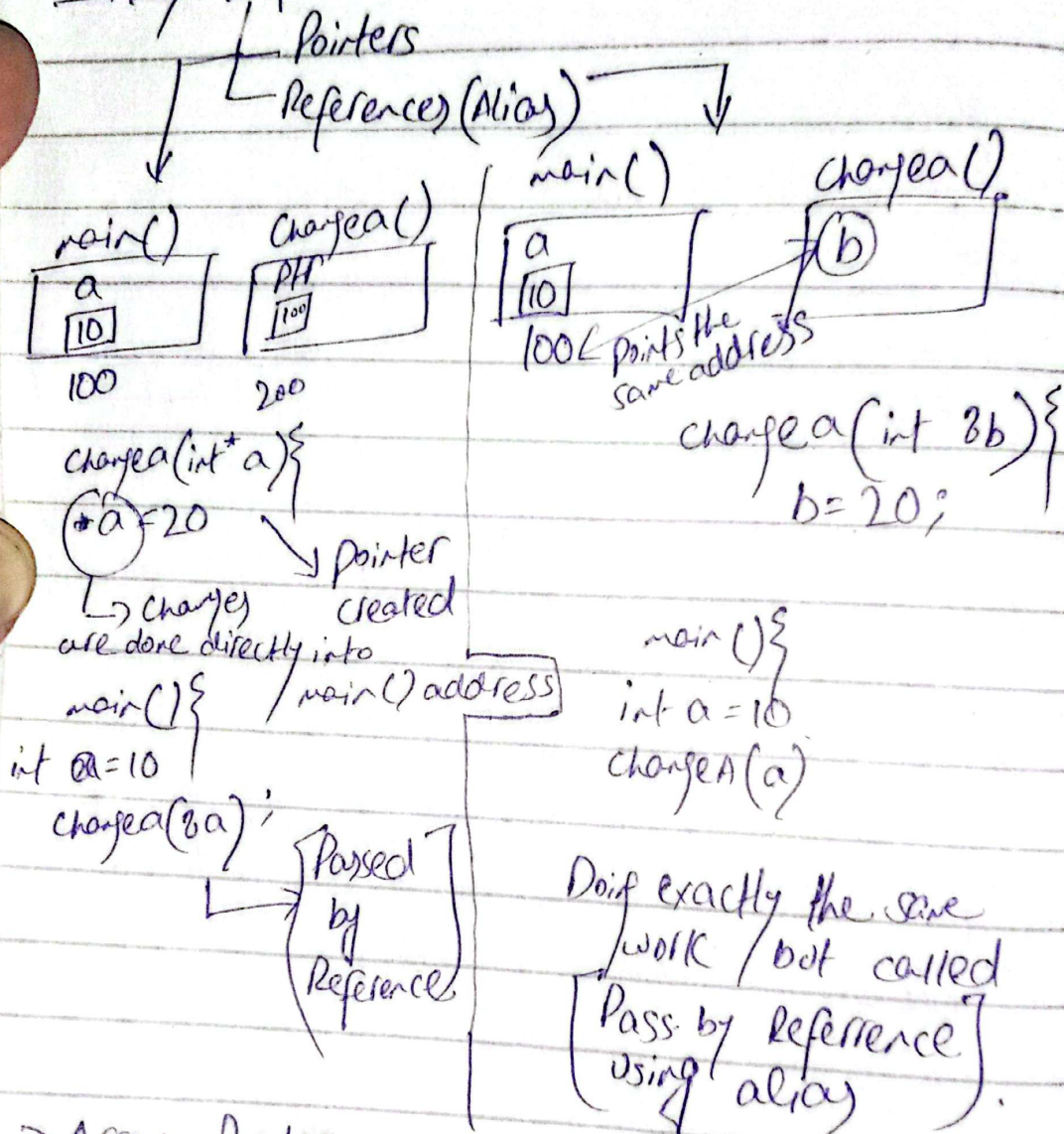
→ NULL Pointer

A pointer that does not point to any valid location.

memory allocated → 0x0

can't be dereferenced.

Pass by Reference



→ Array Pointer

int arr = {1, 2, 3, 4, 5}

1 2 3 4 5

0 1 2 3 4
↑
arr

→ pointer — constant

cout << arr → add

cout << *arr → 1

→ Pointer Arithmetic

① Increment (++) / Decrement (--)

int a = 10

int *ptr = &a

Pointers increment/decrement increments/decrements the space of the variable in the datatype of the variable whatever the datatype may be

a	ptr
10	100
100	300

ptr = 100 ptr++
ptr = 104 ptr++
(Increases the space)

ptr = 100 ptr--
ptr = 96 ptr--
(Decrements the space)

② Add / Subtract

ptr + 1 → 1 int (4 bytes) → addition
ptr + 3 → 3 int (12 bytes)

Similarly for Subtraction
Used (with Arrays like

0	1	2	3	4
1	2	3	4	5

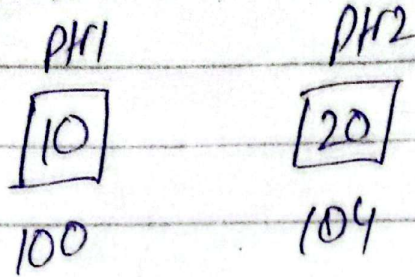
100 104 108 112 116 (addresses)

(arr) → 100 ← → *(arr) → 1
(arr+1) → 104 ← → *(arr+1) → 2
(arr+2) → 108 ← → *(arr+2) → 3
(arr+3) → 112 ← → *(arr+3) → 4
(arr+4) → 116 ← → *(arr+4) → 5

Similarly, for the Subtraction as well.

Compare ($<, >, <=, =, !=$)

Addition of two pointers is not possible
although, subtraction is possible for comparison
purposes



$ptr1 - ptr2 = (4)$ — represents the
4 byte/1 int
difference

All the relational operators
can be applied for the comparison
purposes

$(ptr1 > ptr2)$ or $(ptr1 < ptr2)$
 $(ptr1 >= ptr2)$ or $(ptr1 <= ptr2)$
 $(ptr1 == ptr2)$ or $(ptr1 != ptr2)$

← ————— →
→ Memory Address

→ Pointer ; Pointer to Pointer

→ Dereferencing

→ Pass by Reference

ptr
ref(alias)

* → multiply
→ De-Reference
Operator

→ address of
→ &ref(alias)
→ Pass by
Reference