

Subject \_\_\_\_\_

Date / /

## Memory addresses

way Address will, note A Q8

} = [S] [E] CMMV + N

int age = 30;

{ } → refers to Pointers

printf("%P", age);

↳ to show the memory address

Pointers → type of data

↳ refer to a Physical address inside of the memory

\* We can Create a Pointer Variable to store memory address. (Like we create "char to store character and number")

int age = 30;

int \* PAGE = & age; } or int \* PAGE; }  
PAGE = & age; }

Note: If we do that.

int age = 30;

int \* PAGE;

PAGE = & age;

\* PAGE = 5;

Page will be 5

Like we say  
age = 5;

Page

I have Said

- \* `printf("%P", PAGE);` ✓
- \* `that's correct` ✓
- \* `printf("%P", *PAGE); X`
- (we should say) `printf("%d", *PAGE);`

\* e

جبل الخواجہ

"not in mainline yet" (not in mainline yet)

لهم - مرحوم

لهم اذهب عن ربي و عن عبادك

الله اعلم

## Pointers In Deep

### Types of Pointers

① Integer Pointer ( $\text{int}^*$ )  $\rightarrow$   Addresses

↳ نوعها هو حماية

② address of متغير (++)  $\rightarrow$   Addresses

③ Character Pointer ( $\text{char}^*$ )  $\rightarrow$   Addresses

④ address of متغير (++)  $\leftarrow$   Addresses

$\text{char ch = 'A';}$  /  $\text{char}^* \text{ptr} = &ch;$

⑤ float Pointer ( $\text{float}^*$ ):

$\text{Pi = 3.14;}$  /  $\text{float}^* \text{ptr} = &Pi;$

⑥ Double Pointer ( $\text{double}^*$ ):

: ( ) Like float

but with more precision

⑦ Void Pointer ( $\text{void}^*$ ):

Void Pointer  $\rightarrow$  Base type can

\* Conversion

Cast  $\rightarrow$  لavarz ما قرحة تحويل

## Subject

## موضوع الدرس

Date \_\_\_\_\_

التاريخ

# 2) Null Pointers

Volume 4 Number 9

```
int * Ptr = Null; // Initialize null
```

→ Pointers of Pointer:

int x = 10; } // main function

85

int \*P<sub>s</sub>

$\sin \theta \propto P_s$

$$P = g x_j \quad ?$$

$$PP = \alpha P; \quad , 98 \} S.F \} \rightarrow E)_{\text{MAD}} + N_i$$

آخر Pointer في عنوان لـ  $\text{char}^*$

## 8) Function Pointer

```
Void greet() { cout = "Hello User"; }
    printf("Hello User");
}
```

returning to return

int main {

```
void (* funPtr) () = greet; * fun
```

fun Cptr();

return;

}

$x = x + n$

$q = q + n$

$f = f + n$

## 9) Array Pointer

```
int arr[] = { 1, 2, 3 }; i = 0, j = 4
int * Ptr = arr;
```

$i = 0, j = 4$

$i = 0, j = 4$

## How to read Pointers

Spiral outwards Anti clockwise

start with name of integer

① int \*x [10]

an array of 10 Pointers to integer

② (\*x) [10]

Pointer to an array

(1) First