Subject _____ موضوع الدرس

Date / / التاريخ / /

typecasting

تحويل نوع البيانات مرة واحدة او اكثر

ex int → float

float → int

1) implicit typecasting

يحدث تلقائياً بدون تدخل من المستخدم

2) Explicit typecasting

يتطلب تدخل لتحديد التحويل يدوياً

ex

```
int y=10, x=20.5;
int * ptr = &y;
Float * ptr2 = &x;

ptr2 = (float *) ptr;
```

تشير ال حتواه الى كام ← ptr2

في ptr وطبق البيانات float

أندلسية

مشابهة الاستخدام في

malloc

realloc

Subject _____ موضوع الدرس

Date / / / / التاريخ

Pointers to Pointers.

int *p = &x;

int **p = &p

We can say → *p = p

**p → يخزن عنوانه
تاني Pointer

Arrays vs Pointers.

1D

we know that

int A[10];

int *p;

"A is a Pointer of array"

p = A; OR p = &A[0]

Here we see that A ≡ &A[0]

And Here the Pointer Point to the address of the first num of the array

if we need to reach the next num we urn add 4 bytes ≡ 1 integer

أندلسية

p+1 = A+1 = &A[1]

Subject _____ موضوع الدرس

Date / / التاريخ / /

2D array,

int B[2][3];

→ 00 01 02

→ 10 11 12

A[0] } → 1-Darray of 3 integers
A[1] }

we should say,

int (*P)[3] = B; ✓

P pointer to array that contains 3 integers

B[i][j] = *(*(B+i)+j)

ex. In my P.C

Printf(B) → result

→ *B → B[0] → &B[0][0]

→ B+1 → &B[1]

→ *(B+1) → B[1] → &B[1][0]

→ *(B)+1 → B[0]+1 → &B[0][1]

Subject _____ موضوع الدرس

Date / / التاريخ / /

Pointer to functions

```
int Add (int a, int b) {
    return a+b;
}
int main () {
    int *c (int, int);
```

c = add ✓ → c = &add ✓

printf (*c (2,3)) ✓ → c(2,3) ✓

Pointer as functions

Pointer with back like address

```
int * Add(int *a, int *b){
    int c = *a + *b
        return &c;
    }
int main() {
    int *c
```

c = add(&a, &b);

لـ قيم الاستقبال في Pointer

Subject _____ موضوع الدرس

Date / / التاريخ / /

ex

```
Void A() {
    Printf("Hello");
}
Void B(Void(* Ptr)()) {
    Ptr();}
}

int main()
    Void (*P)() = A ; // we can say P = A

    B = P;

}
```

Subject _____ موضوع الدرس

Date / / / / التاريخ

Little Endian / Big Endian

برامج ترتيب تخزين البايتات Endian

```
int x = 0x12 3456 78;
```

Address	[1]	[2]	[3]	[4]
Value	0x12	0x34	0x56	0x78

Big ~~Little~~ دا برنامج ال

Little ~~Big~~ اصلا ال

[1]	[2]	[3]	[4]
0x78	0x56	0x34	0x12

In "Little" → we start from right
" Big → we start from left as usual to Contact