CS 354 @ Epic - Spring 2019

Lecture 3

(1)

Review

1. Java → C
2. Variables / types
3. Conditionals if, else, else if, switch.
4. loops - while, do...while, for
5. functions.
6. arrays.
7. pointers!

Today

1. Arrays
2. Pointers.
3. 2d arrays.
4. Structures
5. Dynamic memory allocation.

```
int a = 1000;
int *pa = &a;
```

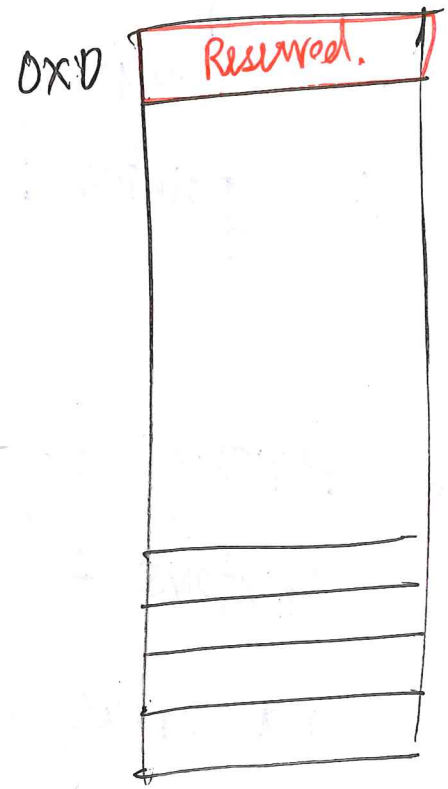
$*pa \Rightarrow *(\&a) \Rightarrow a$

```
int *p = NULL;
```

addr 0 is a special addr. that is reserved by the OS.

```
printf("%d\n", *p);
```

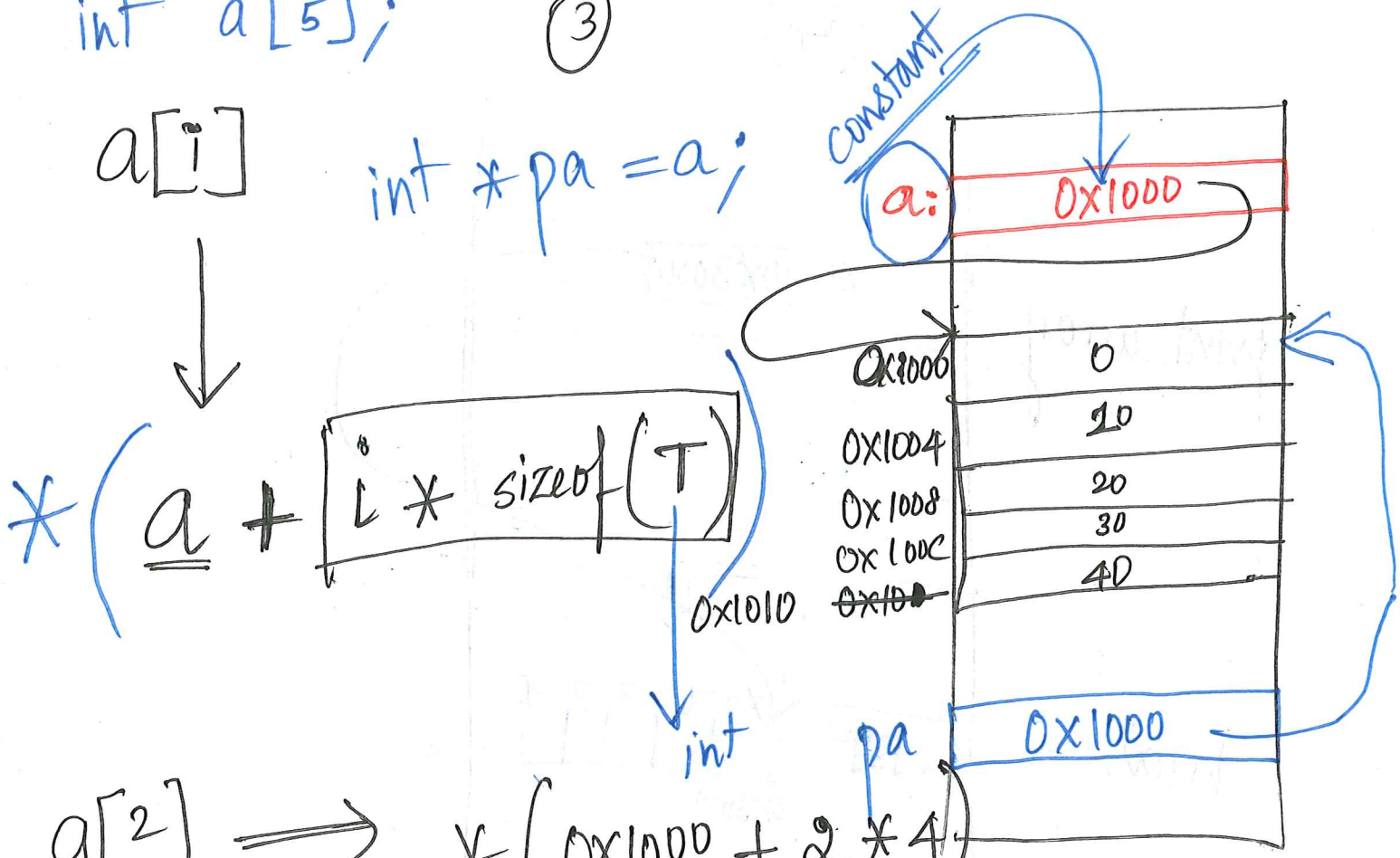
#define NULL 0



int a[5]; ③

a[i]

int *pa = a;

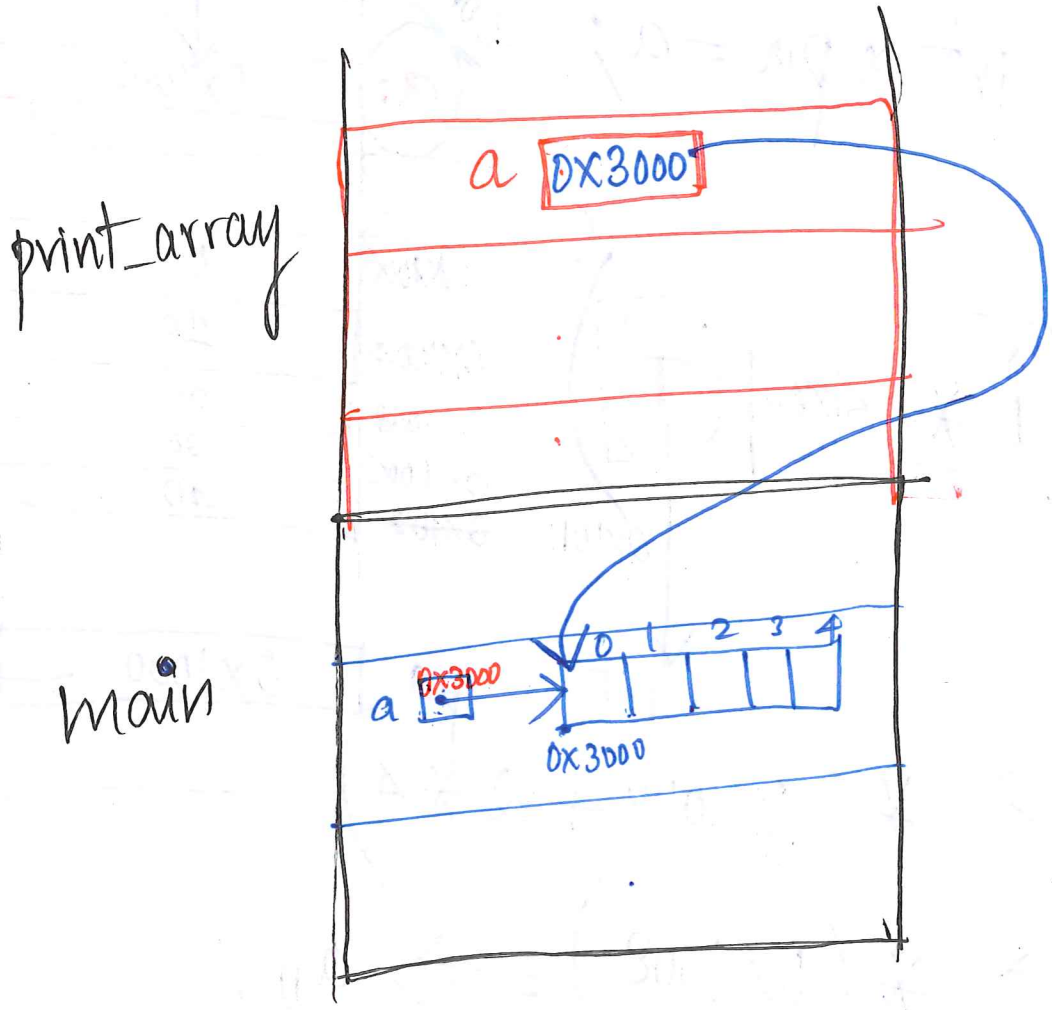


$$*\left(a + \left[i * \text{sizeof}(T)\right]\right)$$

$$a[2] \Longrightarrow *(0x1000 + 2*4)$$

$$\Longrightarrow *(0x1008) \Longrightarrow 20.$$

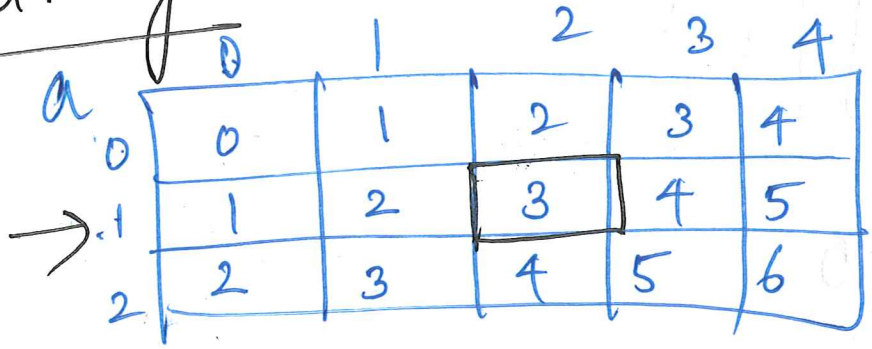
$$a[i] \longleftrightarrow *(a+i)$$



print_array (a);

Arguments are passed by value to functions!

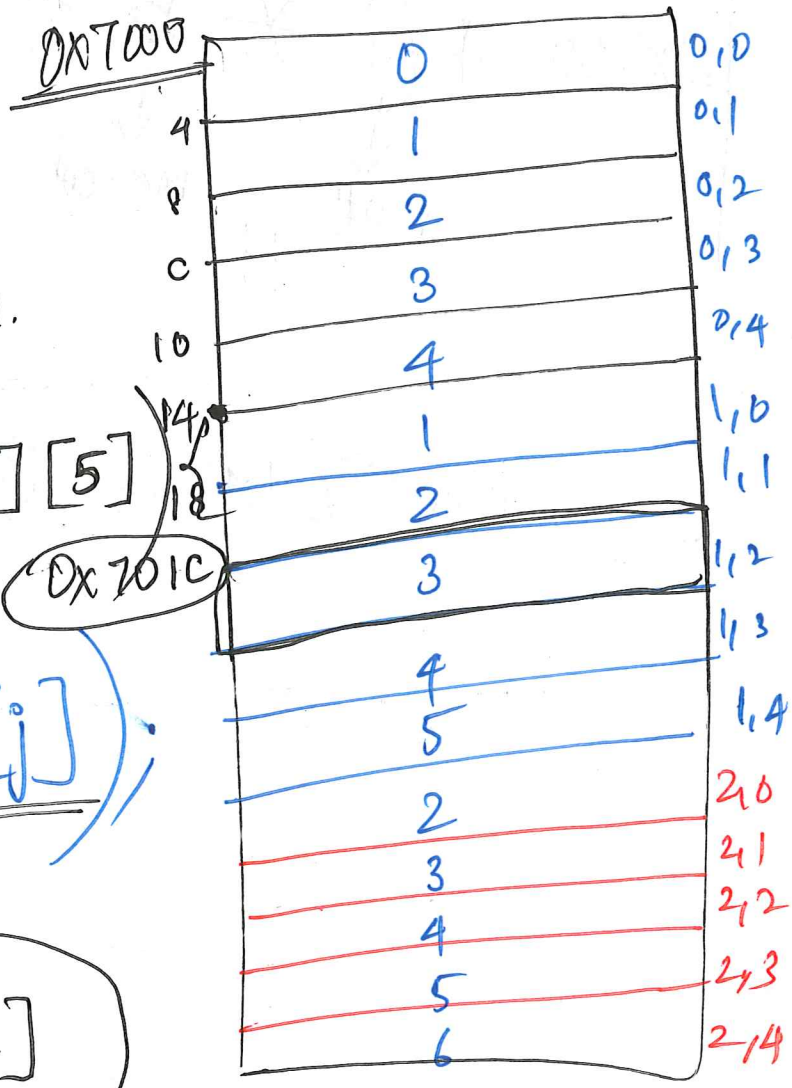
2d arrays




```
int a[3][5];
```

a[i][j]

row column.




```
void print_array(int a[][5])
{
    printf("%d\n", a[i][j]);
}
```

a[1][2]

(6)

a[1][2]

a → 0x7000

max col → 5

$$\left[0x7000 + (1 * 5 * 4)\right] + 2 * 4$$

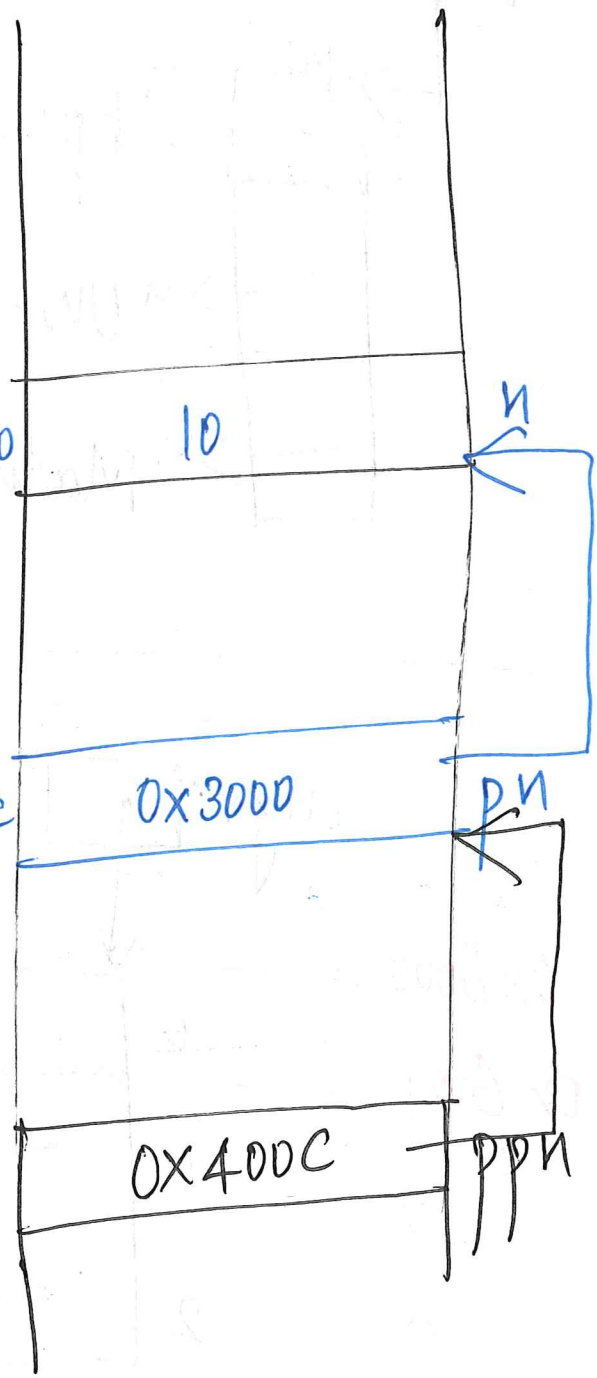
1 → row#, 5 → maxcol, 4 → sizeof(int), 2 → col#, 4 → sizeof(int)

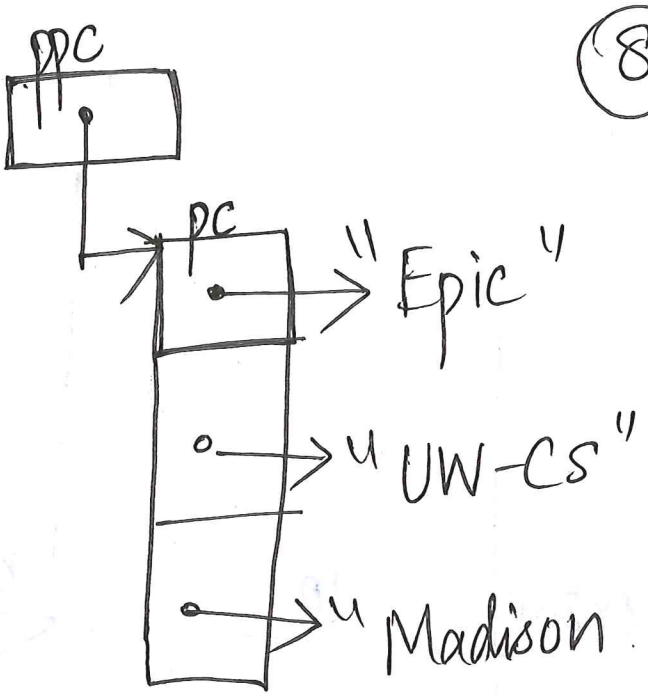
$$= 0x7014 + 8 = \boxed{0x701C}$$

Pointers to a pointer

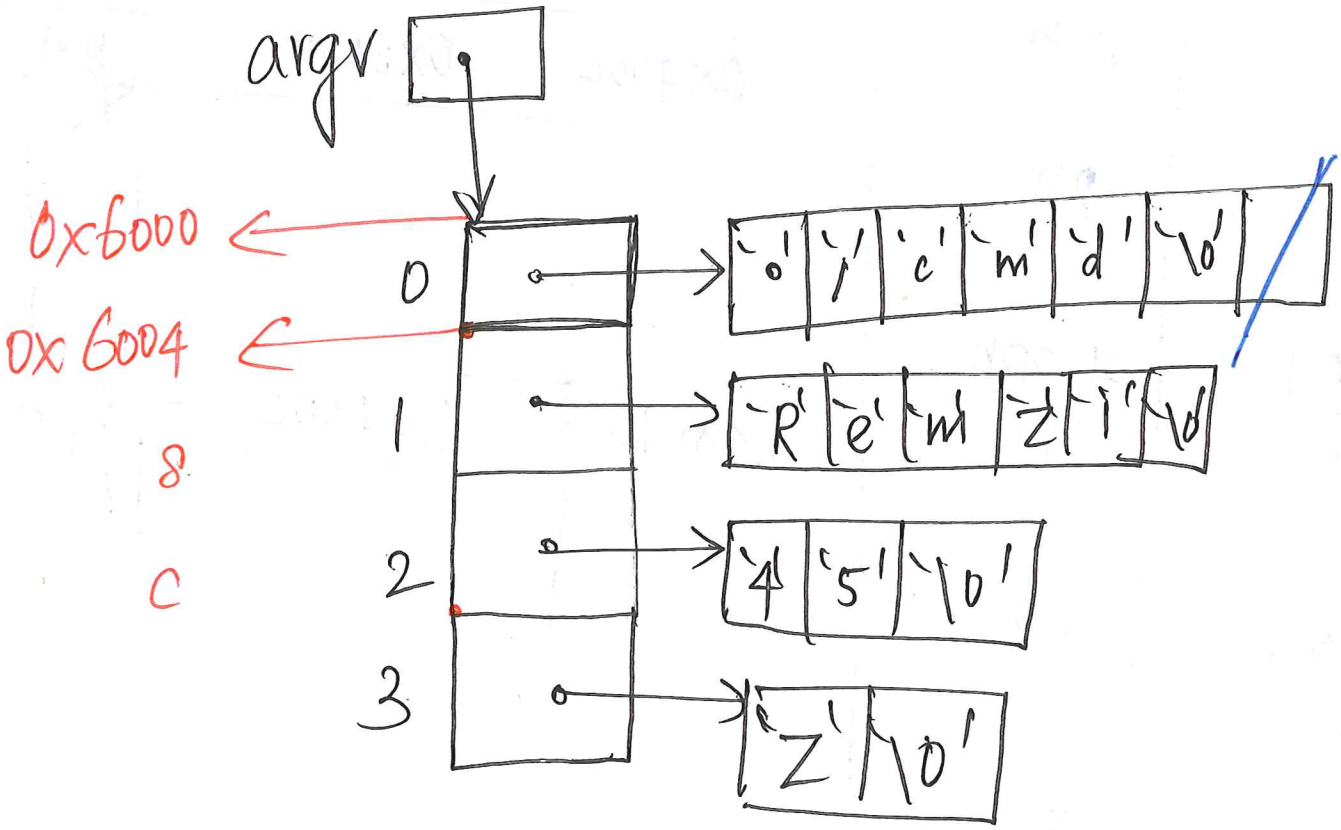
```
int n = 10;
int *pn = &n;
int **ppn = &pn;
```

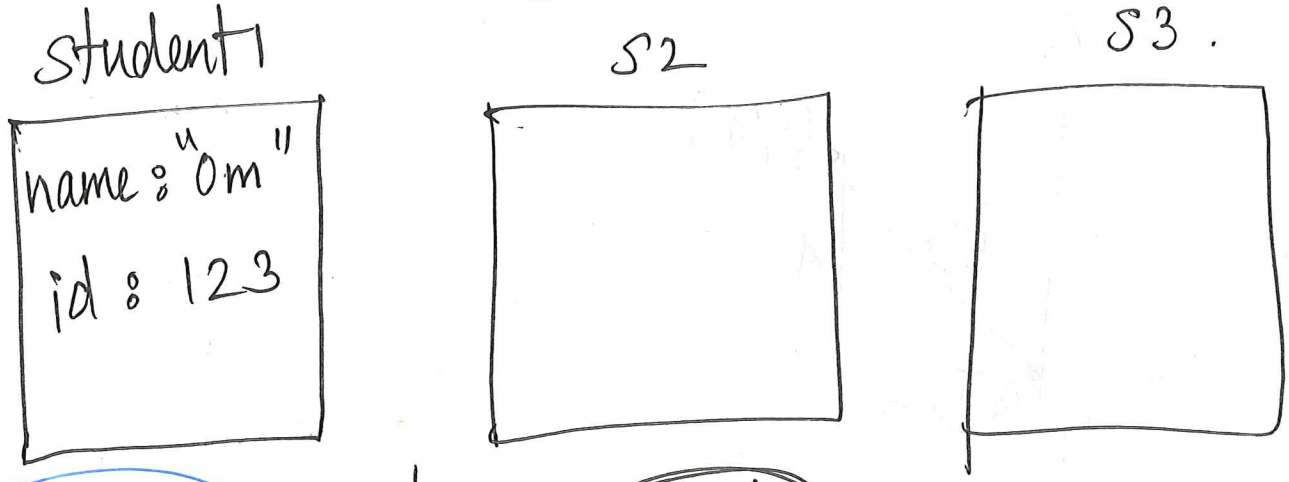
10	0x3000
n	&n
*pn	pn
**ppn	*ppn
0x400C	0x8010
&pn	&ppn
ppn	




ppc
pc
"Epic"
"UW-CS"
"Madison"

char * ppc[3];


argv
0x6000
0x6004
8
c
0
1
2
3
'o' 'i' 'c' 'm' 'd' '\0'
'R' 'e' 'm' 'z' 'i' '\0'
'4' '5' '\0'
'z' '\0'



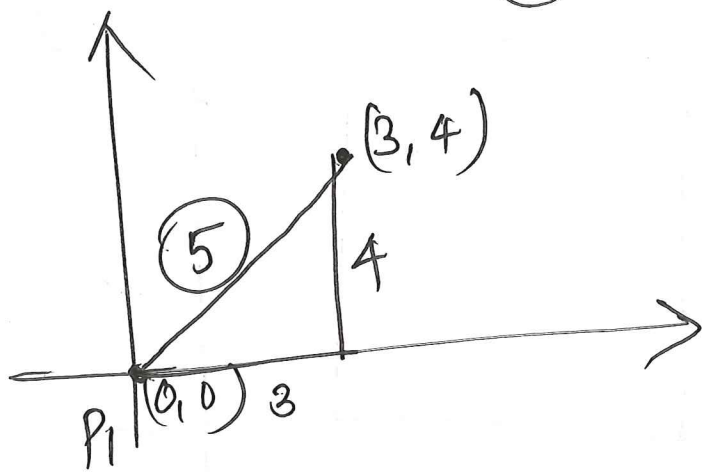
"class"

struct

```
struct student {
    char [] name;
      OR
    char *
    int id;
};
```

```
struct student S1;
int n;
```

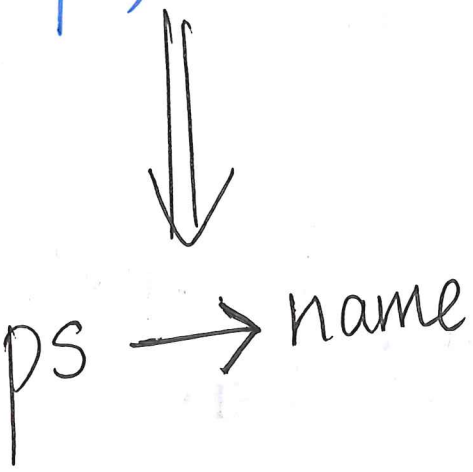
```
S1.name = "Om";
S1.id = 123;
```



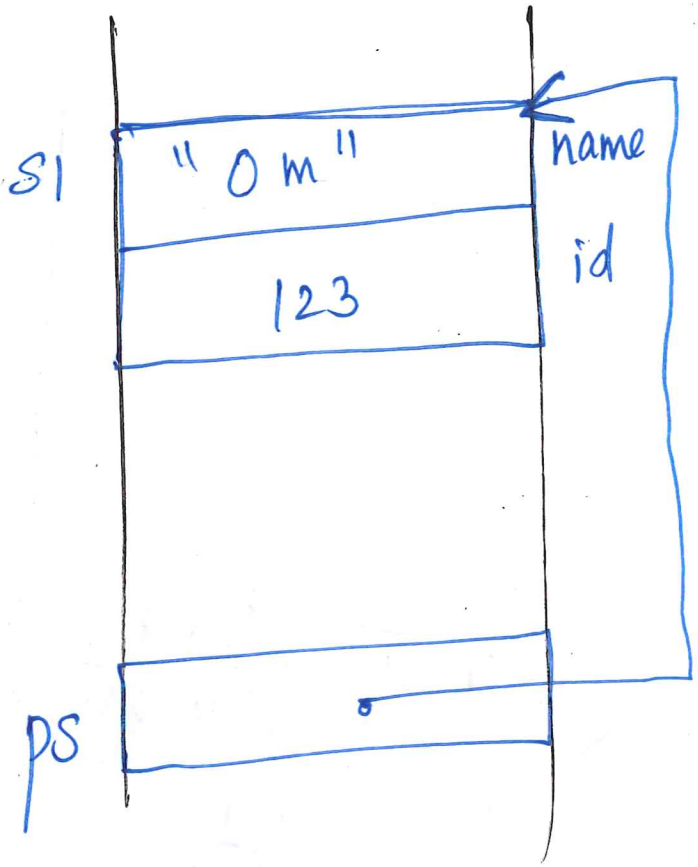
---

struct student * ps = &s1;

(*ps).name

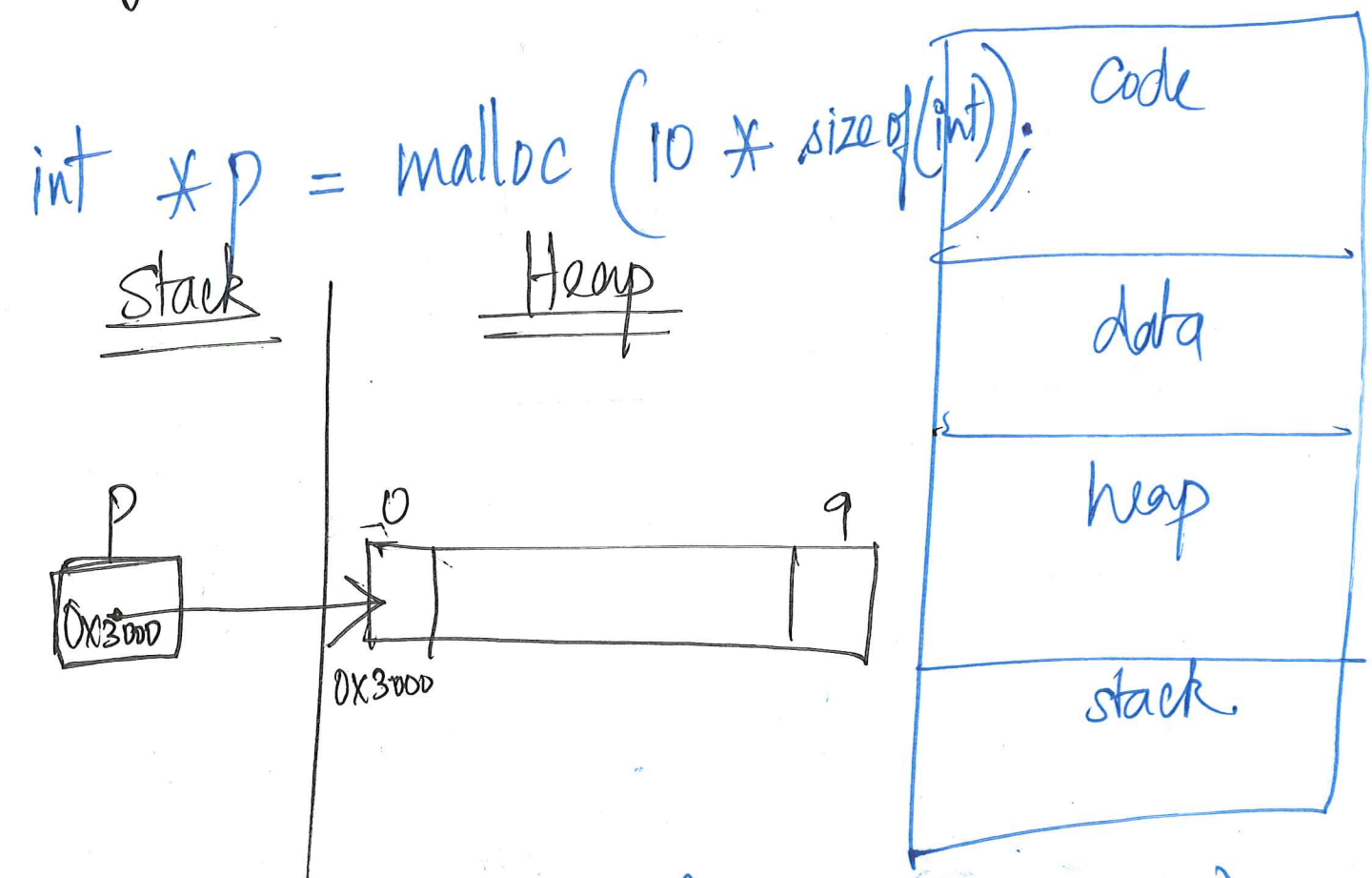


ps → name



Dynamic Memory Allocation.

```
int *p = malloc(10 * sizeof(int));
```




```
int **p = malloc(sizeof(int*) * 10);
p[0] = malloc(10 * sizeof(int));
```