2510 : Lecture 13 Feb 08, 2017

Simulating pass by reference

ex. Want to write a function that triples an int

1 attempt :

```
void triple (int n) {  ①
    n *= 3;  ②
}
```

```
int a = 1;
triple (a);  ③
```

still 1.

/* triple gets a copy of a, changes the copy */

midterm qs

To "pass-by-reference" :

1. put an EXTRA * before the parameter you want to modify

```
void triple (int * n) {
```

2. within function, dereference the parameter to get to the original (follow the arrow or address)

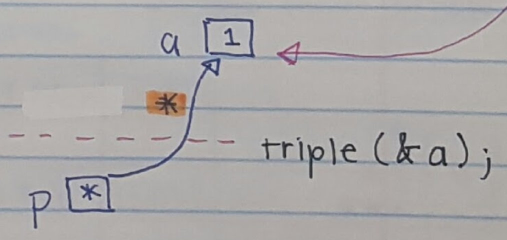
```
*n *= 3;
```

3. when calling the function, pass in the address

```
triple (&a);
```

If we need to change smtg in param, add * and dereference.

```
void triple (int * p) {
    *p *= 3;
}
```



is it clear what's going on:

卌 I

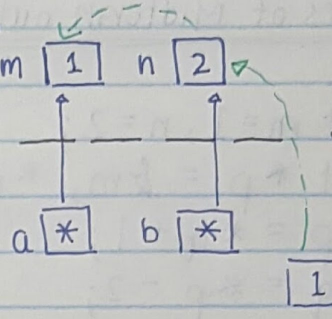
... Lecture 13 ||

int m = 1;
int n = 2;

ex. Function to swap 2 integers

like java, need to assign a temp var.

```
void swap (int *a, int *b) {     step 1
    int temp = * a;
    *a = *b;                      step 2*
    *b = temp;
}
```



step 3
swap(&m, &n);

midterm qs
remember definition of pointers

ex. Function to swap 2 pointers to int

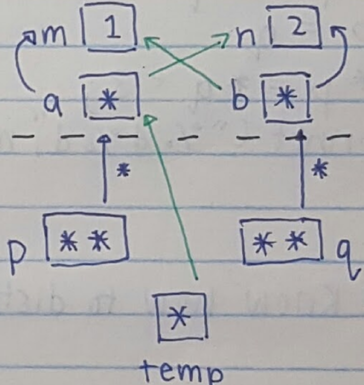
follow steps

1. add * to param
2. add * to inside function
3. pass address with & in swap()

```
int m = 1, n = 2;
int * a = &m, * b = &n;  step 1
void swap(int * * p, int **q) {
    int * temp = * p;
    *p = *q;   step 2
    *q = temp;
```

same as

```
temp = a;
   a = b;
   b = temp;
```



step3
swap(&a, &b);

assignment of pointers

alias!

- given two pointers (of the SAME TYPE) p and q:
  p = q → makes p point to same thing as q

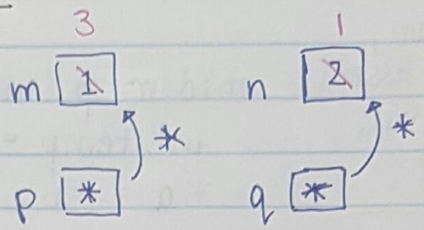
▷ Midterm format

Section 1: writing code
2: write output.

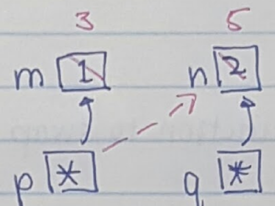
... Lecture 13‖

▸ Examples of Midterm output questions

```
            1) int m = 1, n = 2;
               int *p = &m, *q = &n;
m = n + 1      *p = *q + 1;
n = m - 2      *q = *p - 2;
               printf ("%d%d",m,n);    3 1
```



```
2) int m = 1, n = 2;
   int *p = &m, *q = &n;
   *p = *q + 1;
   p = q
   *p = *q + 3;
   printf ("%d%d", m, n);    3 5
```



!!! Know how to distinguish between p = q and *p = *q