2510 : Lecture 12 || February 6, 2017

Pointers

↳ a pointer is a variable that stores an address.

- 3 questions:
  1) how to declare pointers?
  2) what are some sources of addresses?
  3) what can we do with pointers? (what are its operations?)

Sources of Addresses

2 sources:

1) array names: can be used as starting address of array
2) address of operator (&)

```
p - pointer of an appropriate file
int n = 12;
p = &n;
```

Pointer Declaration

read right to left

```
int *p;   /* p is a pointer to an int */
int n = 12;
p = &n;
```

* = pointer

ie. it stores the address of int var

```
int * p;
```

target type (destination type)

→ the target type is needed because when we go to the destination (specified by the address stored in the pointer) we need to know how to interpret the bit pattern there and how many bytes to interpret

Why do we need target type?

ex. 3700 Ave. building — what type is our destination?

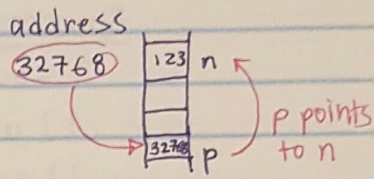
Operations

Dereferencing pointers

p - pointer
*p - dereference p → go to destination address stored in p.

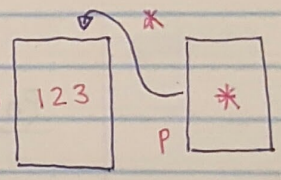
ex.
```
int n = 123;
int *p = &n;
printf("%d", *p); /* 123 */
*p = 456; /* change n to 456 */
printf("%d", n); /* 456 */
```

does this change value of n? indirectly.



Definition: We say that the pointer p points to x if p contains the address of x.

```
int n = 123;
int *p = &n;   ← p points to n.
*p = 456;
int **q = &p;
```



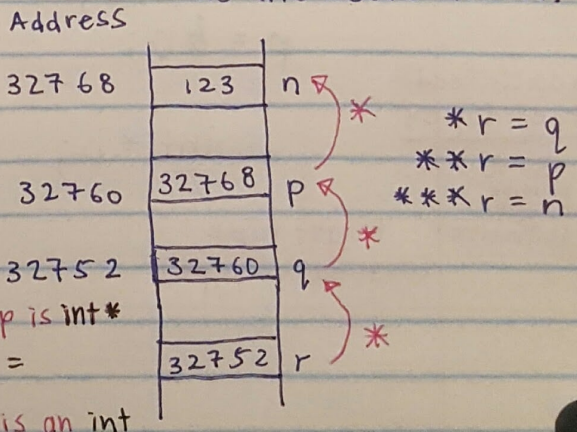
Think of the * operator as the above arrow.

we could also create a variable

double pointer to int

```
int ***r = &q;
```

triple pointers and onwards are barely used.




```
int n = 123;
int *p = &n;
```
p is int*
p is an int

```
int *q = &p;
```
**q is an int

```
int **q
```
q is an int**
q is an int*

cont.

... Lecture 12 ||

- a pointer gives us an indirect way to get to something

  ∴ If a function takes an address, it can use it to get to an outside object to change it.