this is week 4

fall 2013

playlist50
Stay Stay Stay (Taylor Swift)
Little Wrecking Ball (Miley Cyrus)
Counting Stars (OneRepublic)

agenda

resources

pointers

memory pt. 1

SPL

resources

lecture notes & source code

cs50.net/shorts

study.cs50.net

man

Google

cs50.net/discuss

OHs

me!

Quiz 0

Wed 10/16 @ 1:00 PM

covers weeks 0-5

review session next Mon

review++ next Tues in section

resources

// logic

draw a picture

write some pseudocode

// syntax

map it onto C

code the program

http://sayat.me/cs50

+ hands on (pseudo)coding
Δ wait longer for questions

pointers

0x7734

pointers

data stored in memory has
both a *value* and an *address*

int x = 50;

x

```
50
```
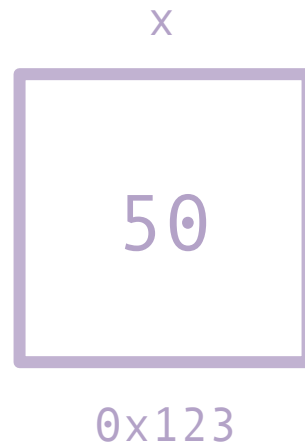
0x123

pointers

a pointer's value *is* an

address

int x = 50;   int* ptr = &x;

x

```
┌──────┐
│      │
│  50  │
│      │
└──────┘
```

0x123

ptr
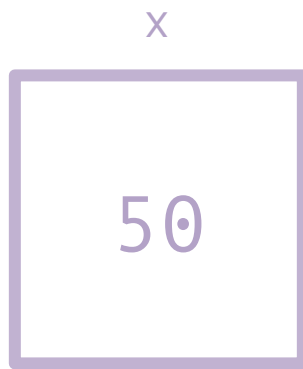
```
┌──────┐
│      │
│0x123 │
│      │
└──────┘
```
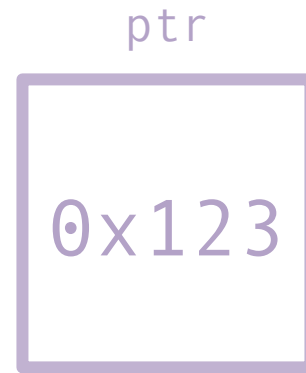
0x456

pointers

```c
// declare a pointer
int* ptr;
// get an address and store it in a
pointer
int x = 50;
ptr = &x;
// go to an address and get its value
// *ptr is equivalent to x
printf("%i\n", *ptr);
```

your turn: pointers.c

Sketch out on paper what the following code does and then check your logic with pointers.c.

```
int x = 2, y = 8, z = 12;          z = x * y;

    int* ptr_x = &x;                   x *= y;

    int* ptr_y = &y;                y = *ptr_x;

    int* ptr_z = &z;              *ptr_x = x * y;

                                  ptr_x = ptr_y;

                              x = (*ptr_y) * (*ptr_z);
```

arrays

...are pointers!

char array[6];

F	O	P	S	C	\0
array[0] *array	array[1] *(array + 1)	array[2] *(array + 2)	array[3] *(array + 3)	array[4] *(array + 4)	array[5] *(array + 5)

your turn: trainer.c

Write a program that takes one and only one command-line argument: the number of dolphins to be trained (> 0). Allow the trainer to enter an age for each dolphin (> 0). To get an age, call a function with the following prototype:

`int* getAge(void);`

All ages should be stored in an array in main. Finally, print out the age of the oldest dolphin.

Please start with some pseudocode...

your turn: trainer.c

validate if input was given and is > 0

initialize an array to save dolphin ages

for each dolphin

    getAge()

find the largest integer

    print it out

for each dolphin

    get an integer

    validate if input is > 0

your turn: trainer.c

condition   validate if input was given and is > 0

         initialize an array to save dolphin ages

loop   for each dolphin

        getAge()

loop   find the largest integer

printf   print it out

loop   for each dolphin

GetInt   get an integer

condition   validate if input is > 0

memory pt. 1

stack attack

stack

a part of memory

local variables stored in stack "frame"

variables go away when function returns

dynamically allocated memory

```
void* malloc(int number_of_bytes)
sizeof(data_type)
void free (void* pointer)


// reserve enough space to store 6 chars
int* ptr = malloc(6 * sizeof(char));


// free this space
free(ptr);
```

SPL

what's cookin' good lookin'?

SPL

```
// create an object
GRect paddle = newGRect(x, y, width, height);
```

SPL

```
// do something with an object
<function>(<where>, <what>);
setColor(paddle, "BLACK");
add(window, paddle);
```