



→ clicker.cs.illinois.edu

code: 340 ←

CS 340

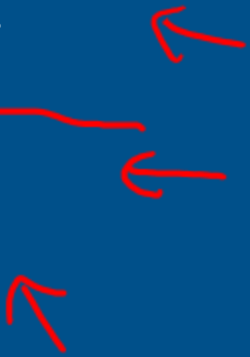


QR code coming soon

C without the ++

slides on website / examples

Learning Goals

- Meet your classmates/people you can study with!
 - Start to understand why we are using C.
 - Be able to compare C++ code to C code.
 - Be able to read and understand C code.
 - Be able to write C code from scratch.
- 

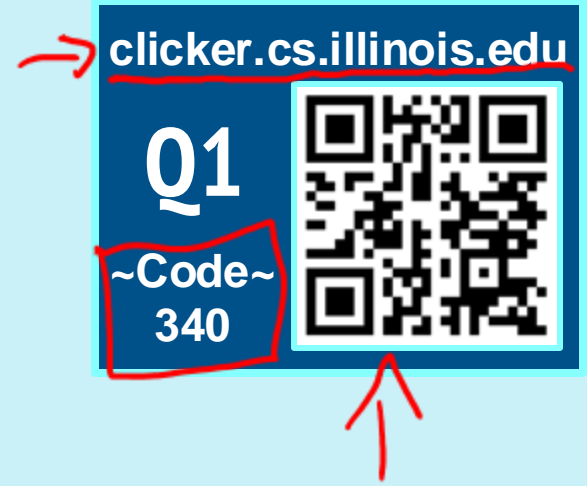
Plan for Today

- Listen and participate during a presentation about C.
 - Clicker questions!
- Follow along as I go through a coding demo.
- Work with a partner on your coding in C assignment.

To get 0.1% extra credit today you need to

- Answer a majority of the clicker questions
- Stay until the end of class

What is your experience with C (C++ and Tuesday don't count)?



Ice breaker!

20%

- Introduce yourself to a classmate
 - Name
 - Major
 - If they have ever coded in C before
- Clicker: What percentage of the class do you think has never done any C coding?

clicker.cs.illinois.edu

Q2

~Code~
340



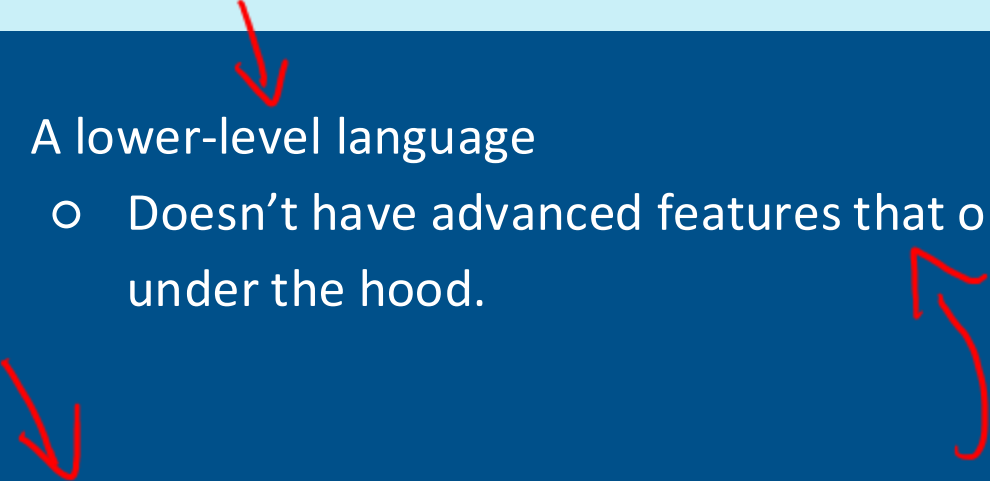
50%

100%





Why C?

Where did the ++ go...?

- A lower-level language
 - Doesn't have advanced features that obscure what is happening under the hood.
 - C is used in future courses that this class is a preq for.
- 

How is C different from C++?

What does the ++ mean?

- No templates
- No classes (only basic structs)
- No overloading functions or operators
- No new or delete (instead malloc and free)
- No pass-by-reference
- No standard C++ library 
 - No strings, vectors, maps, or lists 
 - No cout or cin

How confused/nervous are you to code in C without the ++ features?

Not at
all

Very!



How to use malloc/free
instead of new/delete

Malloc / Free

Using dynamic memory

```
//allocates size bytes on the heap and returns a  
//pointer to that memory location on the heap.
```

```
void *malloc(size_t size);
```



```
//frees the memory at ptr from the heap
```

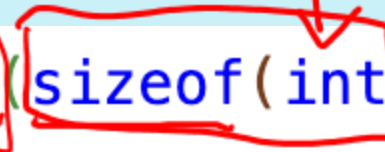
```
void free(void *ptr);
```



Example

~~int ptr~~

```
36  → int *ptr_int = malloc(sizeof(int));
37  → *ptr_int = 10;
38  → free(ptr_int);
```



4



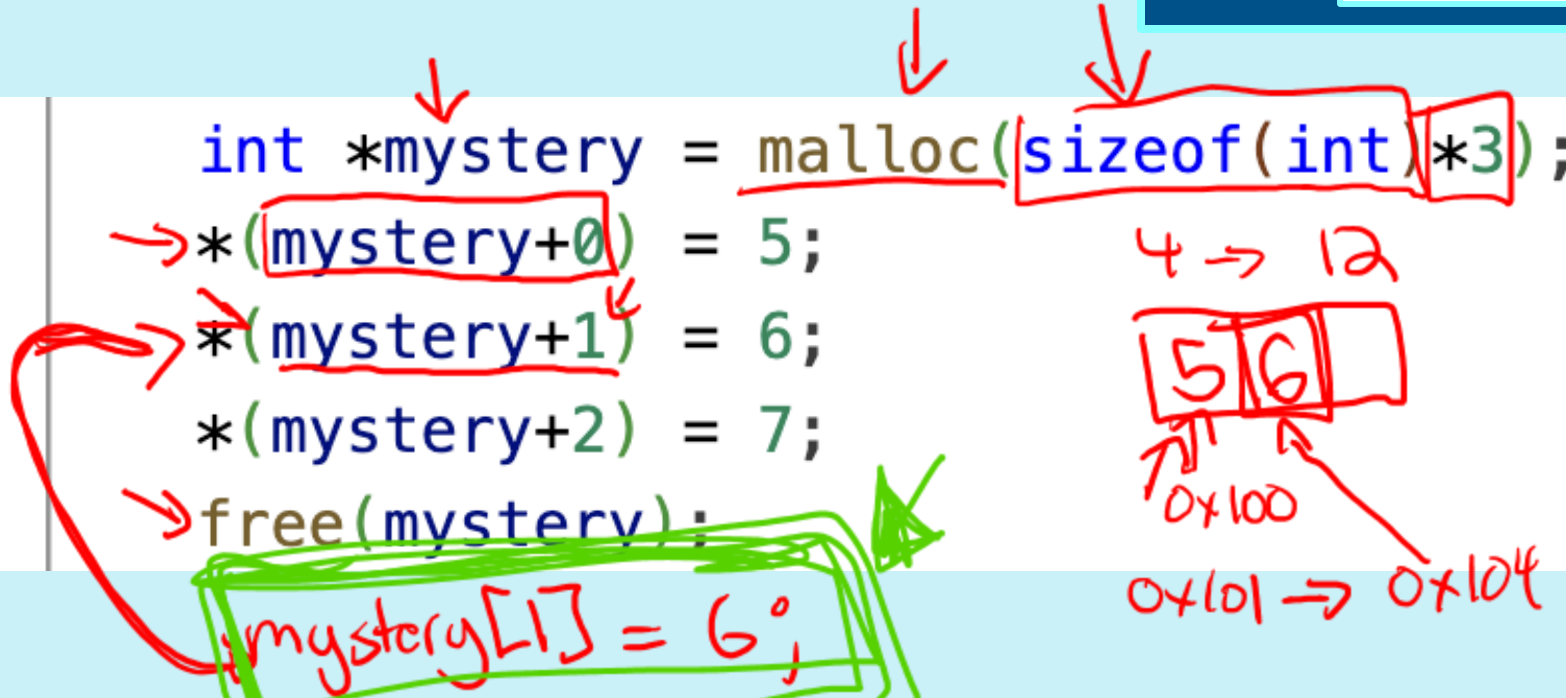
Q3

~Code~
340



What is an equivalent line 43?

```
41     int *mystery = malloc(sizeof(int)*3);
42     *(mystery+0) = 5;
43     *(mystery+1) = 6;
44     *(mystery+2) = 7;
45     free(mystery);
```

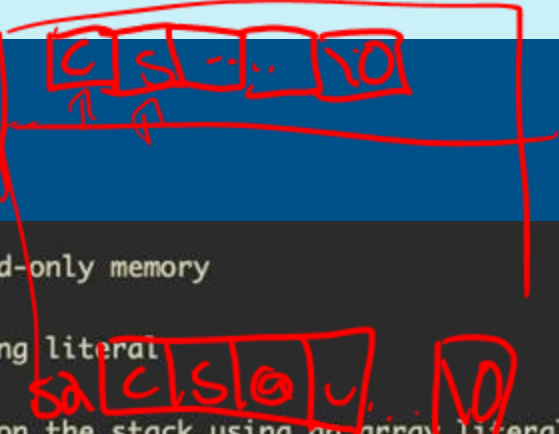


`mystery[1] = 6;`

How to deal with
sequences of character
in C with no `std::string`

C-Strings

```
char* s = "CS@UIUC"; // Set a char pointer to point to string literal in read-only memory  
char s2[8] = "CS@UIUC"; // Initialize a char array on the stack using a string literal  
char s3[8] = {'C', 'S', '@', 'U', 'I', 'U', 'C', '\0'}; // Initialize a char array on the stack using an array literal  
char* s4 = malloc(8); // Dynamically allocate memory for a string then write a string literal to that memory  
strcpy(s4, "CS@UIUC");
```



0	1	2	3	4	5	6	7
C	S	@	U	I	U	C	\0

~~free(s4)~~
~~s[0] = '\0'~~
s2[0] = '\0'



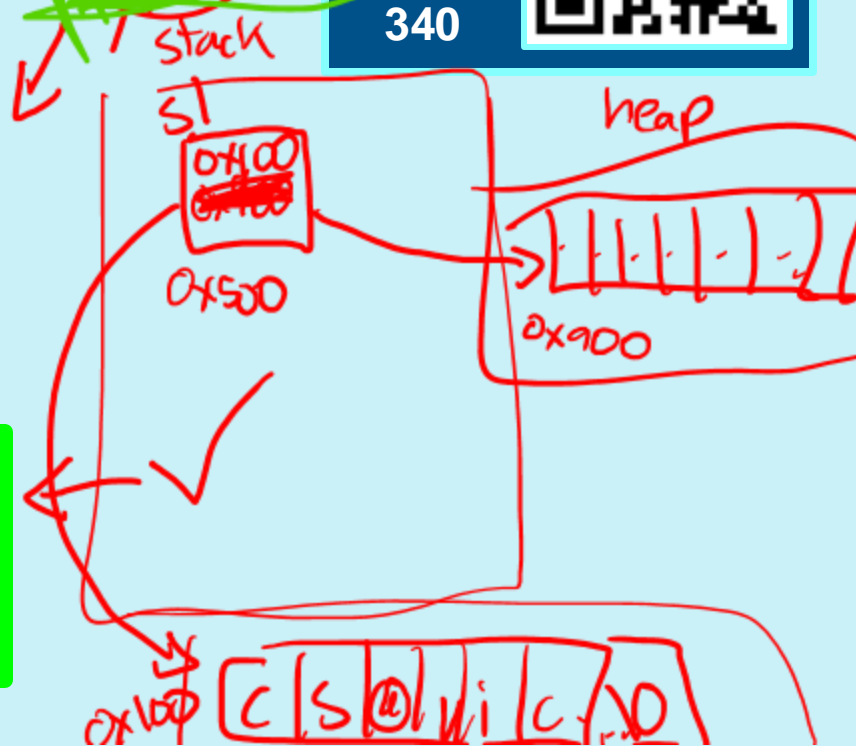
What is the error in the top version?

memory error and memory leak

```
40 char *s1 = malloc(8);  
41 s1 = "CS@UIUC";  
42 free(s1);
```

s1[0] = 'C' ...

```
34 char *s2 = malloc(8);  
35 strcpy(s2, "CS@UIUC");  
36 free(s2);
```



How to work with user-
defined objects if you
cannot have
classes/member
functions

structs

No member functions or operator overloading

Strategy/style

1. Create a struct with the member variables.
2. Write functions that take in a pointer to an instance of that struct and use that pointer to access and manipulate the member variables.

C Demo

An example similar to your homework!

Available on Prairie Learn to follow along

What is coming up

- Finish HW (due before class next Thursday (12:30pm))
- Work on MP 0 (due 11:59pm Tuesday)
- Read website text for more details and information
- Tuesday's Topic: CPUs