**#0** – What are the three errors?​

**#1** – When mysteryAnimal is declared, the next instruction prints out: “The animal is initialized to: …” It is never initialized to a variable. Why does this not crash?? What is it being initialized to??​

**It is being initialized to Cat. First Print Line States that. Pointer not initialized. Random Memory Behavior.  
Fixed by initializing pointer to an actual object created.   
Initialized to “unknown” (default) as expected.**

**#2** – Why is your program crashing on this line: “std::cout << "The animal should initally be nothing:…”?​

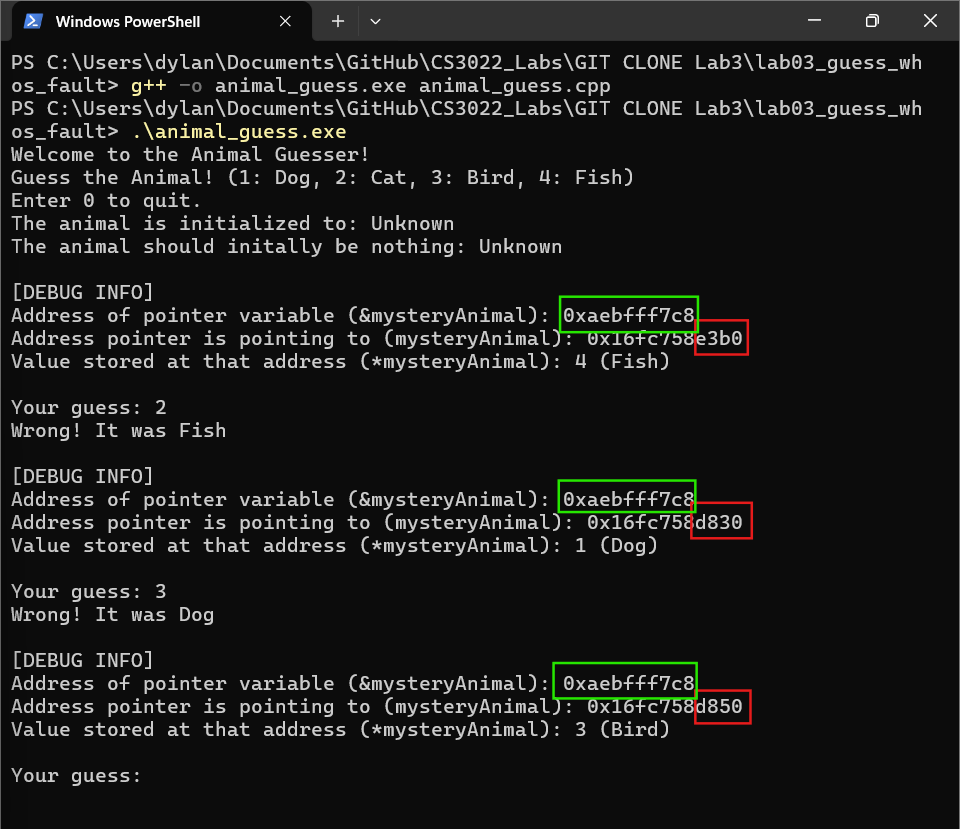
**Comment out line 38**. **Initializing Animal\* to “nullpointer” does bad stuff.   
If it works, don’t ask too many questions.   
Basically, the nullptr removes what it points to, which has all the info. You can’t pull data from something with no data. So it crashes.**

**#3** – Run the program for 3 different guesses. On each guess, print out:​

* - The address of the mysteryAnimal pointer​
* - The address where the pointer is pointing​
* - The value located at the address where the pointer is pointing.​
* - Draw a picture diagraming each of these memory locations with their values.​
* - Is the stack growing from the bottom (0x0000….) or the top (0xFFFFF…) of memory?​

**#4 -**Relative to the stack and heap, where is the staticWelcomeMessage located? What is its address? Add it to the diagram.​ **Address: 0x7ff7404250a0**

**Fix: Put the delete function at the end, but INSIDE the while loop. Not outside. Deletes each mystery animal before making the new one.**

Same address of pointer variable (same pointer)  
Different address of variable pointed to (memory allocation).  
 e3b0 – 58288  
 d830 – 55344  
 d850 – 55376

Grows from the top. (Depends how you draw it, I guess). You place an object, then place another one on top, then another, like a stack of pancakes. Memory is allocated descending order.

Stack Memory (grows downward)

┌───────────────────────────────┐

│ ... other local vars ... │

├─────────────────────────────┤

│ &mysteryAnimal = 0x61FF0C │ ← address of the pointer variable

│ mysteryAnimal = 0x1C6A510 - - - - ┼─────────────┐

├─────────────────────────────┤ │

│ return addr, saved regs ... │ │

└──────────────────────────────┘ │

▼

Heap Memory (grows upward)

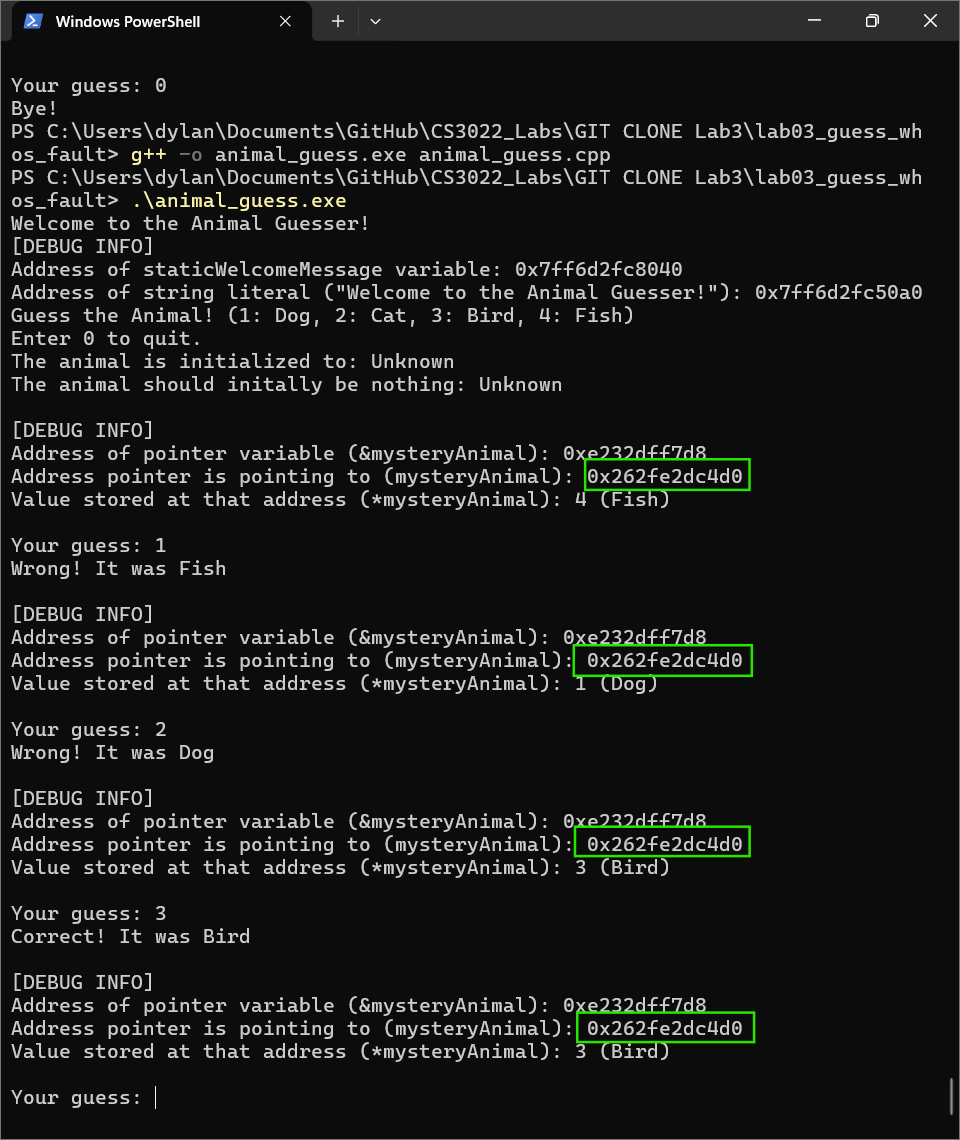
┌───────────────────────────────┐

│ [0x1C6A510]: value 2 ("Cat") │ ← allocated Animal object(s)

├──────────────────────────────┤

│ Global / Static variables │ ← staticWelcomeMessage

└──────────────────────────────┘

**FIX**!  
  
After moving delete to inside the loop, all pointer addresses match. Stack is not growing.  
Problem