

First name (color-in initial)	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	section (9,10,11, 12,1 or 2)	first name initial	last name initial
Last name (color-in initial)	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z			

H10: Due Monday, 11.17 in Lecture

Pointers (Sect. 9.1)

Assigned: Wed 11.12

Total Points: 50

MAY ONLY BE TURNED IN IN THE LECTURE/LAB LISTED ABOVE AS THE DUE DATE, or offered in person, for in person grading, during instructor or TAs office hours. See the course syllabus at <https://foo.cs.ucsb.edu/16wiki/index.php/F14:Syllabus> for more details.

(1) (10 pts) Fill in the information below. Also, fill in the A-Z header by

- **coloring in** the first letter of your first and last name (as it appears in Gauchospace),
- writing **either 9,10,11,12,1 or 2** to indicate your **discussion section (lab)** meeting time
- writing your **first and last initial** in large capital letters.

All of this helps us to manage the avalanche of paper that results from the daily homework.

name:	
uemail address:	@uemail.ucsb.edu

If you collaborated with AT MOST one other person on this homework, write his/her name below. She/he should also have your name on his/her paper.

Reading: Read Section 9.1, pp. 508-520. (If you don't have a copy of the textbook yet, there is one on reserve at the library.)

Then, answer the following questions. Be sure to check both sides.

2. (5 pts) According to our textbook, the term "dynamic variables" refers to variables created with what C++ operator?
3. (5 pts) Our book refers to a special area of memory called the **freestore**. You will typically hear this are called **the heap** by most of your instructors at UCSB. What does your book say that the freestore (heap) is used for?
4. (3 pts) What operator returns unused memory to the freestore (heap)?
5. (3 pts) What C++ unary operator is the "de-referencing" operator?
6. (3 pts) What C++ unary operator is the "address-of" operator?
7. Your textbook describes the ordinary local variables that are declared inside functions, and distinguishes them from dynamic variables—while noting that, possibly confusingly, they are NOT called static variables. (That's something else entirely!) You will learn that these variables are stored "on the stack".

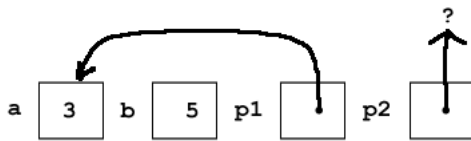
a. (3 pts) What other name, besides "ordinary variables" does our author mention for these variables?

b. (3 pts) These variable get created (on the stack, as we will learn) when a function is called. When are they destroyed? {Space!4em}}

8. Consider the following code:

```
int a=3, b=5, *p=&a, *p2;
```

We can draw a diagram that shows the effect of this code in memory, like this.



Note that we are showing the contents of a and b, and we are showing where p1 and p2 point. Since p2 is uninitialized, we show it pointing to a question mark (?). If a regular int variable were uninitialized, we would show a question mark (?) inside the box for the variable.

Your job is to draw similar diagrams for the code below. I encourage you to do this assignment in pencil in case you need to erase, but in case you don't have one handy, I've given you two boxes for each problem—one for a "draft version" and another for your final "good version". If your draft is "good enough", you don't need to recopy it, but if it is messy, please recopy to a "nice" version.

a. (5 pts)

```
int a=6, b=7, *p1=&b, *p2=&a;
p1 = p2;
(*p1) = 8;
p2 = &b;
```

Scratch work

Final answer

b. (5 pts)

```
int a=2, b, *p1=&b, *p2=&a, *p3;
p3 = p2;
(*p1) = 8;
p2 = p1;
p1 = p3;
(*p2) = 4;
```

Scratch work

Final answer

c. (5 pts)

```
int a=2, b=3, *p1, *p2;
p2 = &a;
p1 = &b;
(*p1) = (*p1) + (*p2);
```

Scratch work

Final answer