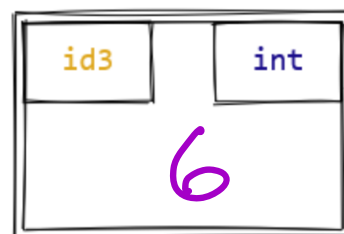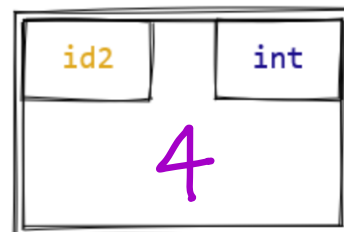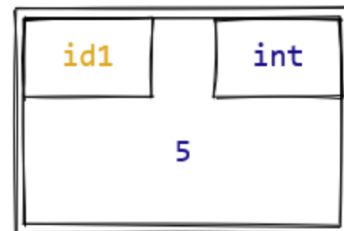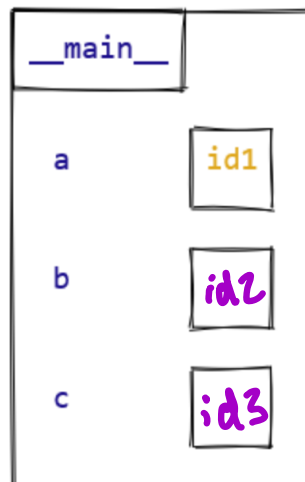# CSC110 Lecture 15: The Python Memory Model

🖨 Print this handout

Exercise 1: Practice with the Python memory model

For each of the following code snippets, fill in the memory model diagram on the right to reflect the state memory after the code is executed. In each case, we begin with the state where a has already been assigne the value 5. You may not need to use all of the boxes.

```
# Question 1
a = 5
b = 4
c = 6
```

__main__

a   id1

b   id2

c   id3

id1		int
	5	

id2		int
	4	

id3		int
	6	

```
# Question 2
a = 5
b = a + 3
c = [a, a, a]
```

__main__

a	id1
b	id2
c	id3

id1	int
	5

id2	int
	8

id3		list
0	1	2
id1	id1	id1
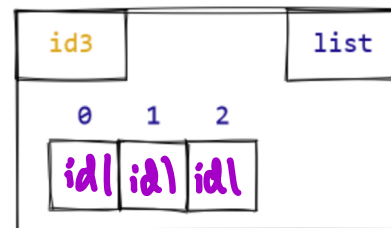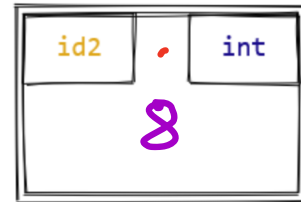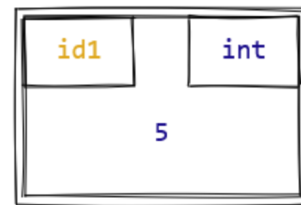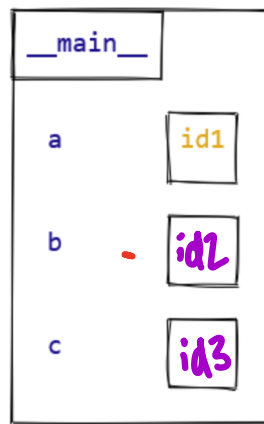
```
# Question 3
a = 5
b = [[a]]
```
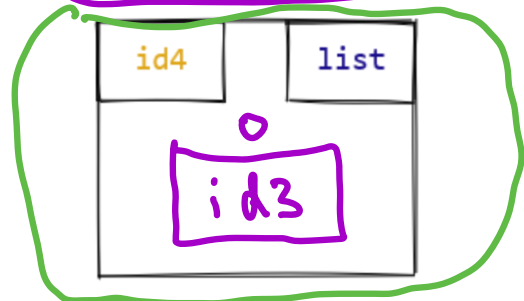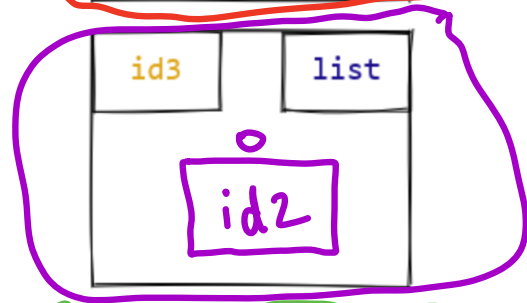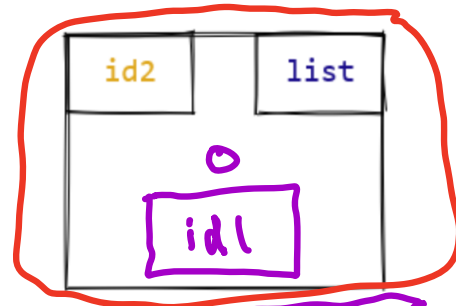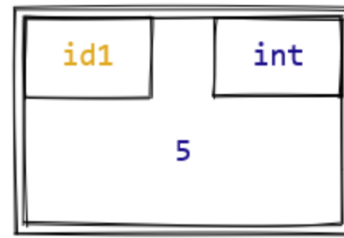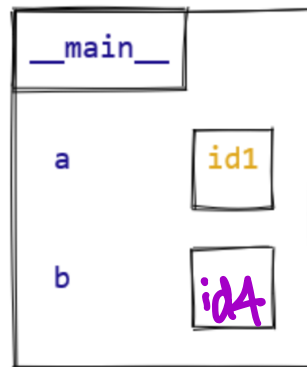

Exercise 2: Variable reassignment and mutation with the memory model

1. Suppose we have two variables `a` and `b` that have been assigned `int` values:

a. Write a snippet of code to swap which values a and b refer to. After your statements are executed, a should refer to the object that b used to refer to, and b should refer to the object tha a used to refer to.

$$t = a$$
$$a = b$$
$$b = t$$

$$a = b$$
$$b = a$$
wrong!
code

Use a third "temporary" variable to perform the swap.

b. Complete the following memory model diagram to show the state of memory *after* your code snippet executes.

$t = a$
$a = b$
$b = t$

__main__

a   id20

b   id 10

t   id10

id10   int

id20   int
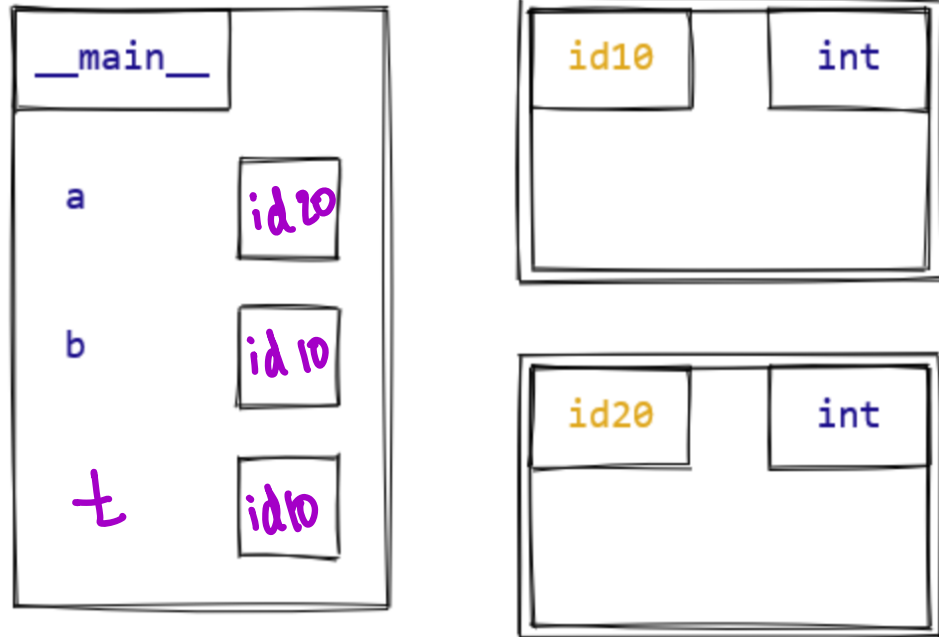
c. Would the code you wrote in part (a) behave correctly if a and b referred to mutable values like lists instead of ints?

yes!

2. Farmer Mario is the proud owner of 57 chickens, 18 goats, and 21 truffle hogs. He is very organized and keeps a log of how many animals of each type he has by the end of each month.

a. Complete the following memory model diagram to represent a variable mario_log to refer to this data, using an appropriate Python data type:

__main__

mario_log   id42

id11	str
"chickens"	

id13	str
"truffle hogs"	

id15	int
18	

id12	str
"goats"	

id14	int
57	

id16	int
21	

id42   dict

id11: id14
id12: id15
id13: id16
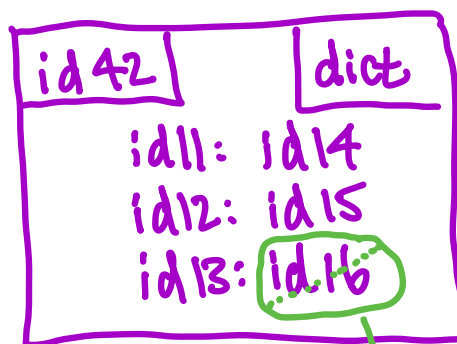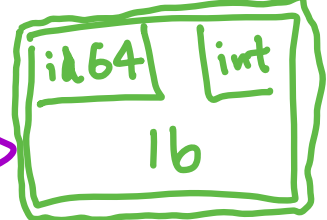
b. At the end of the month, Mario sells 5 truffle hogs to David. Write a line of code to mutate `mario_log` to reflect this change.

mario_log['truffle hogs'] += -5 (handwritten)

[handwritten annotation top right:] id64 | int | 16

c. Modify your above diagram to show this change.

[handwritten:] : id64

Exercise 3: Aliasing

For each snippet of code below, complete the memory model diagram and answer the question below it. (You may need to indicate elements being added/removed from a collection data type.)
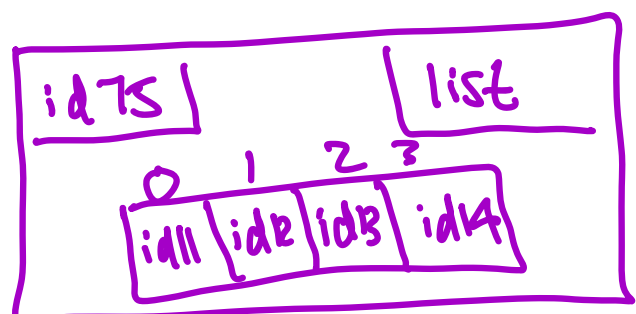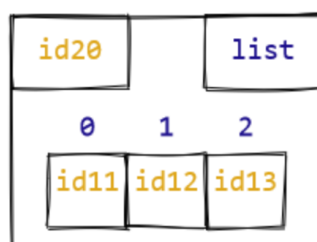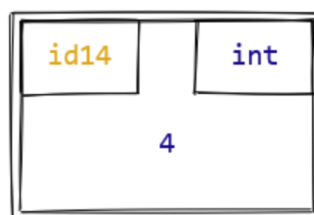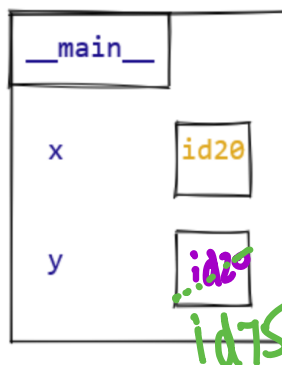
1.
```
x = [1, 2, 3]
y = x
y = y + [4]
```

[diagram:]

__main__
x   id20
y   id20 (crossed out) id75

id11 | int
1

id13 | int
3

id12 | int
2

id14 | int
4

id20 | list
0   1   2
id11 id12 id13

[handwritten box:] id75 | list
0   1   2   3
id11 id12 id13 id14

After the code above is executed, which of the following expressions evaluate to True? Circle those expression(s).

```
a. x == [1, 2, 3] and y == [1, 2, 3, 4]
b. x == [1, 2, 3, 4] and y == [1, 2, 3, 4]
c. x == [1, 2, 3] and y == [1, 2, 3]
d. x is y
```
(a. circled)

2. Consider this code

```
x = {1, 2, 3}
y = x
set.add(y, 4)
```


After the code above is executed, which of the following expressions evaluate to True? Circle those expression(s).

```
a. x == {1, 2, 3} and y == {1, 2, 3, 4}
b. x == {1, 2, 3, 4} and y == {1, 2, 3, 4}
c. x == {1, 2, 3} and y == {1, 2, 3}
d. x is y
```
(b. circled)

3.
```
lst = [[1, 2], [3, 4]]
for item in lst:
    list.append(item, 88)
```

worksheet typos. (handwritten annotation)

__main__

lst	id20
item	id12

id10 (circled handwritten annotation)

id20 / id10 — list

0	1
id11	id12

id11 — list

0	1	2
id20	id21	id24

id12 — list

0	1	2
id22	id23	id24

id20 — int

1

id21 — int

2

id22 — int
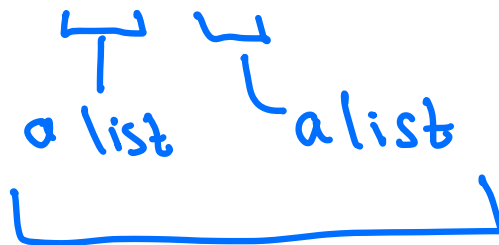
3

id23 — int

4

id24 — int

88

What is the value of lst at the end of this code snippet? $[[1,2,88],[3,4,88]]$ (handwritten)

4.
```
lst = [[1, 2], [3, 4]]
for item in lst:
    item = item + [88]
```

a list   *a list* (handwritten annotations pointing to item and [88])

+
gives a new list with 3 elements that item is reassigned to refer to. (handwritten annotation)

— typos (handwritten, red)

__main__ box:
```
__main__
          id10  (handwritten red, circled; "id10" also written above circled in red)
lst       
item      id97 (purple)
```

id20 list
```
  0    1
id11  id12
```

id11 list
```
  0    1
id20  id21
```

id12 list
```
  0    1
id22  id23
```

id20 int
```
    1
```

id21 int
```
    2
```

id22 int
```
    3
```

id23 int
```
    4
```

id24 int
```
    88
```

id97 list (handwritten, purple)
```
id21  id23  id24
```

[I got confused in class and suggested my solution was incorrect, but it is correct!] (handwritten, purple/red)

What is the value of `lst` at the end of this code snippet?

[[1,2], [3,4]] (handwritten, purple)

Exercise 4: Functions and the call stack

For each snippet of code below, draw the memory model diagram for the state of the program *immediate* *before* the function (`f1`/`f2`) returns. Then, write what would be displayed in the Python console.
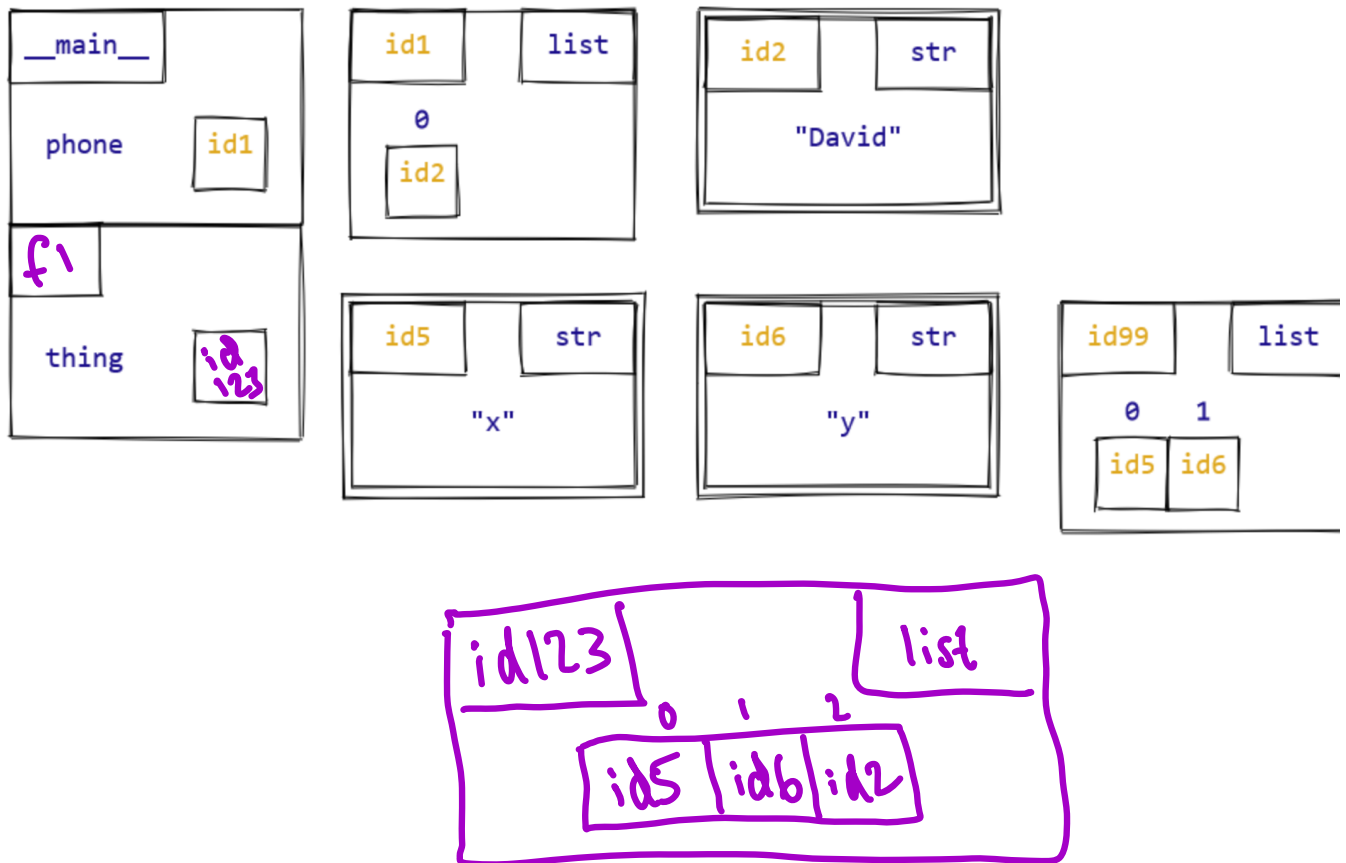
1.
```python
def f1(thing: list[str]) -> None:
    thing = ['x', 'y'] + thing



# In Python console
>>> phone = ['David']
>>> f1(phone)
>>> phone
```

← draw model at this point. (handwritten, red)

['David'] (handwritten, purple)

```
__main__

phone   id1

f1

thing   id123
```

```
id1        list

0
id2
```

```
id2         str

"David"
```

```
id5         str

"x"
```

```
id6         str

"y"
```

```
id99              list

0    1
id5  id6
```

(handwritten model, purple)
```
id123                list

      0   1   2
    id5 id6 id2
```

2.
```
def f2(thing: list[str]) -> None:
    thing.extend(['x', 'y'])
```
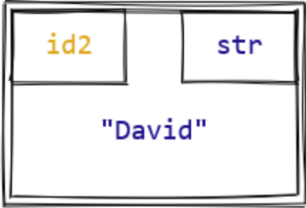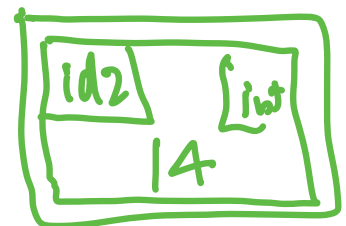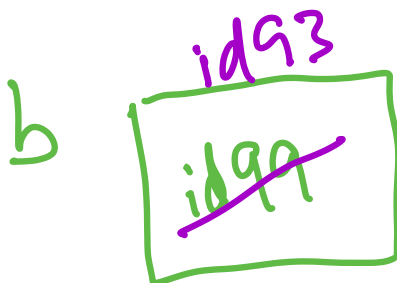
← draw model at this point

```
# In Python console
>>> phone = ['David']
>>> f2(phone)
>>> phone
```
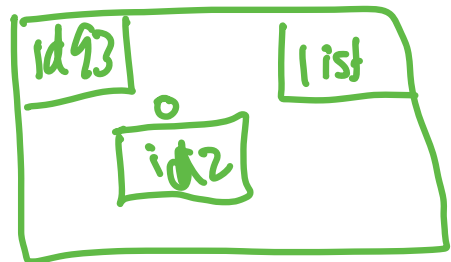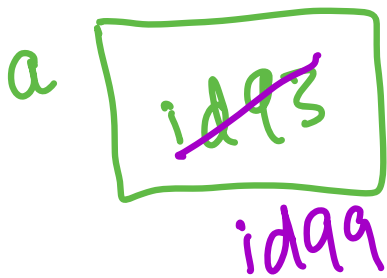
['David', 'x', 'y']

```
__main__

phone      id1

f2

thing      idl
```

```
id1            list

    0     1     2

   id2   id5  id6
```

```
id2            str

      "David"
```

```
id5            str

       "x"
```

```
id6            str

       "y"
```

```
id99           list

    0     1

   id5   id6
```

a ~~id93~~ id99

b id93 ~~id99~~

t = a

a = b

t id93

id93   list
       id2

id2 list
     14

id99   list
       id | id
       27   30

id30 list

b = t

x = [1, 2, 3]

y = [1, 2, 3]

z = x

x = [4, 5, 6]

x | id10 | id12

y | id11

z | id10

id10				list
	0	1	2	
	id1	id2	id3	

id11				list
	0	1	2	
	id1	id2	id3	

id12				list
	id4	id5	id6	

id1 — 1
id2 — 2
id3 — 3
id4 — 4
id5 — 5
id6 — 6