

# CSC110 Lecture 14: The Python Memory Model

Hisbaan Noorani

October 13, 2020

## Contents

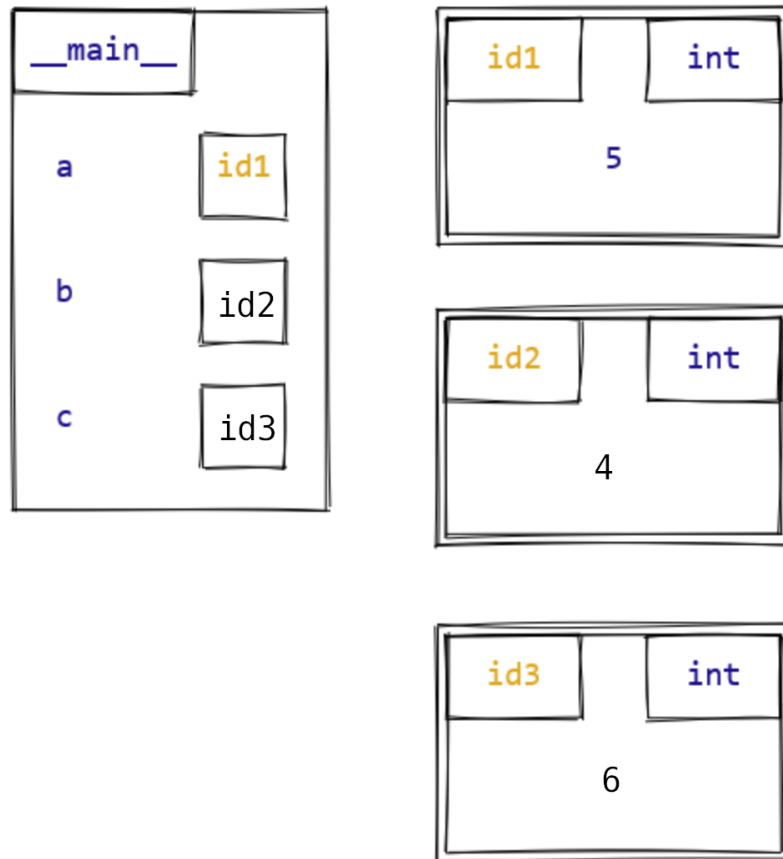
<b>1</b>	<b>Exercise 1: Practice with the Python memory model</b>	<b>1</b>
<b>2</b>	<b>Exercise 2: Variable reassignment and mutation with the memory model</b>	<b>5</b>
<b>3</b>	<b>Exercise 3: Aliasing</b>	<b>6</b>
<b>4</b>	<b>Exercise 4: Functions and the call stack</b>	<b>10</b>

## **1 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 of memory after the code is executed. In each case, we begin with the state where `a` has already been assigned the value 5. You may not need to use all of the boxes.

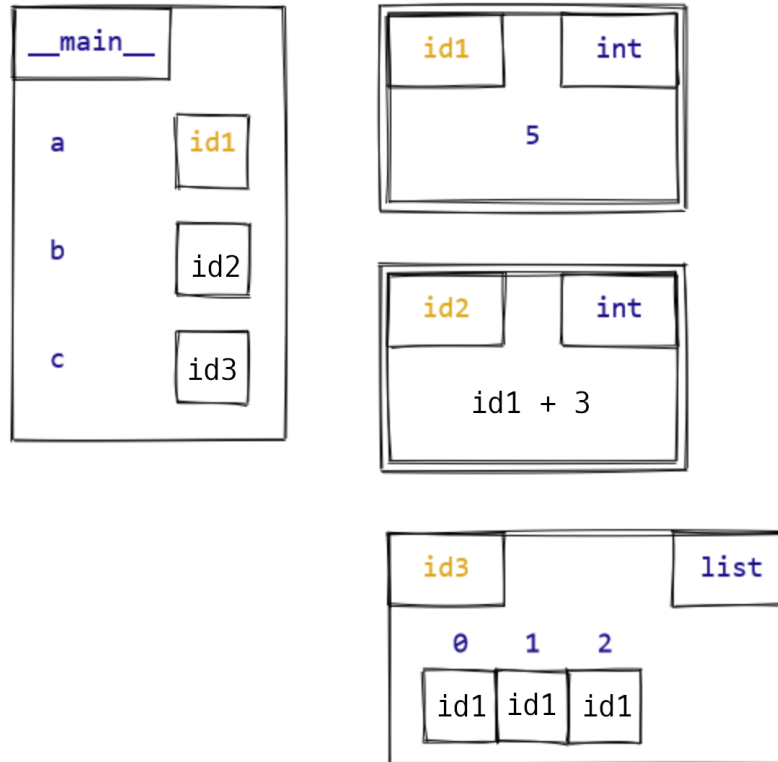
1. .

```
1 a = 5
2 b = 4
3 c = 6
```



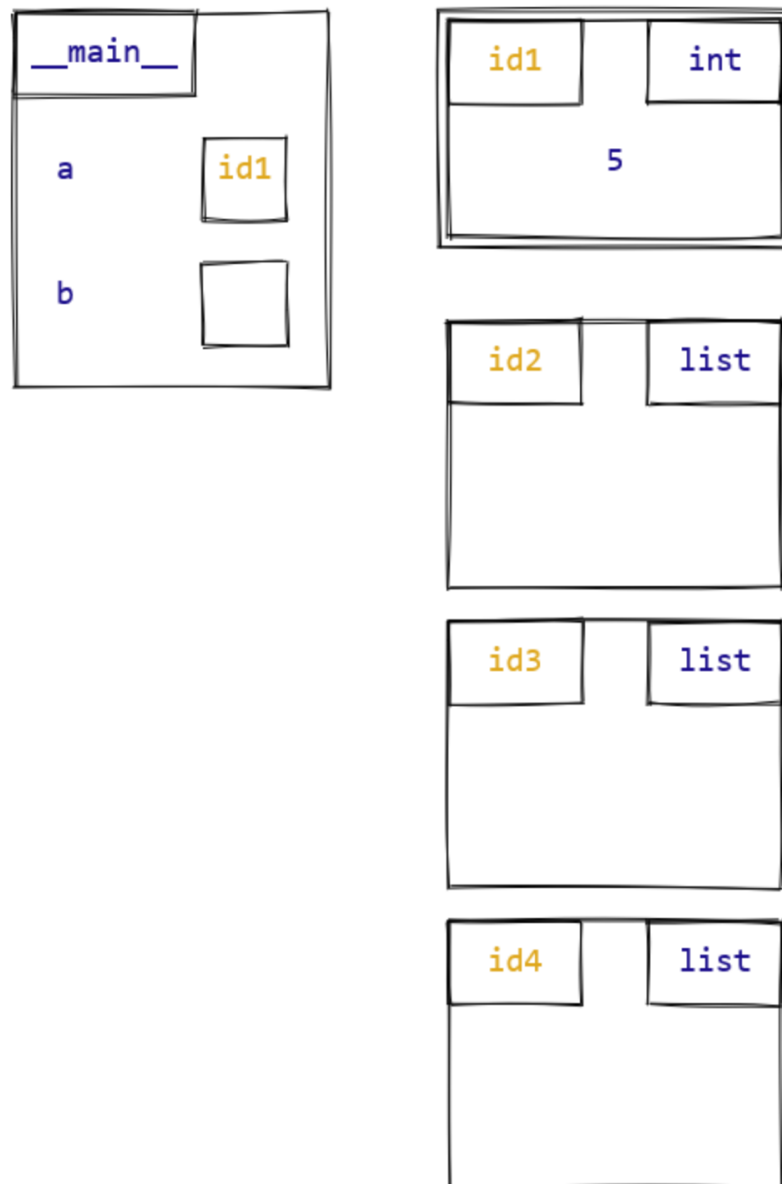
2. .

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



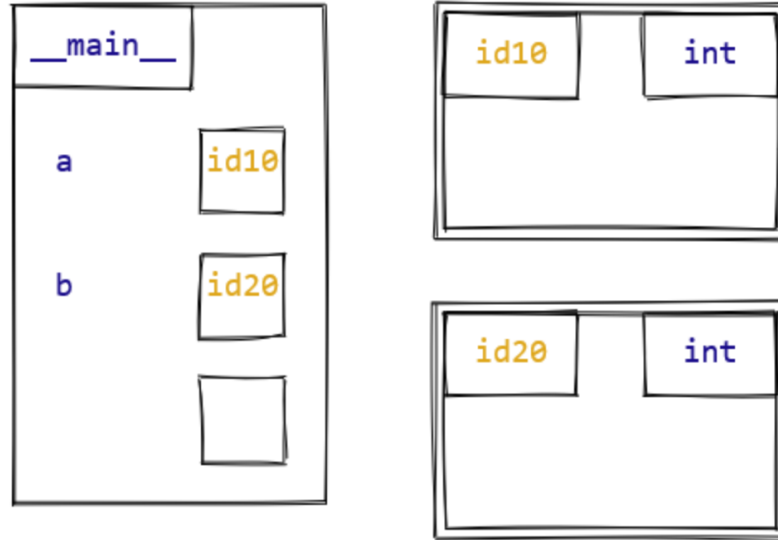
3. .

```
1 a = 5
2 b = [a]
```



## 2 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:

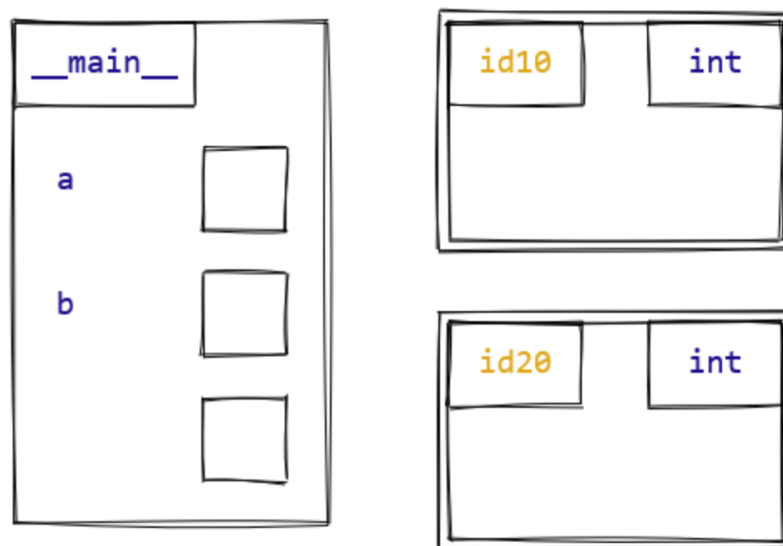


- (a) Write a snippet of code to swap which values `a` and `b` refer to. After your statements are executed, `a` should refer to the object that `b` used to refer to, and `b` should refer to the object that `a` used to refer to.

Use a third “temporary” variable to perform the swap.

```
1 a = temp
2 a = b
3 b = temp
```

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



- (c) Would the code you wrote in part (a) if `a` and `b` referred to mutable values like `~list~`s instead of `~int~`s?
1. 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:



- (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.
- (c) Modify your above diagram to show this change.

### 3 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. .

```
1 x = [1, 2, 3]
2 y = x
3 y = y + [4]
```

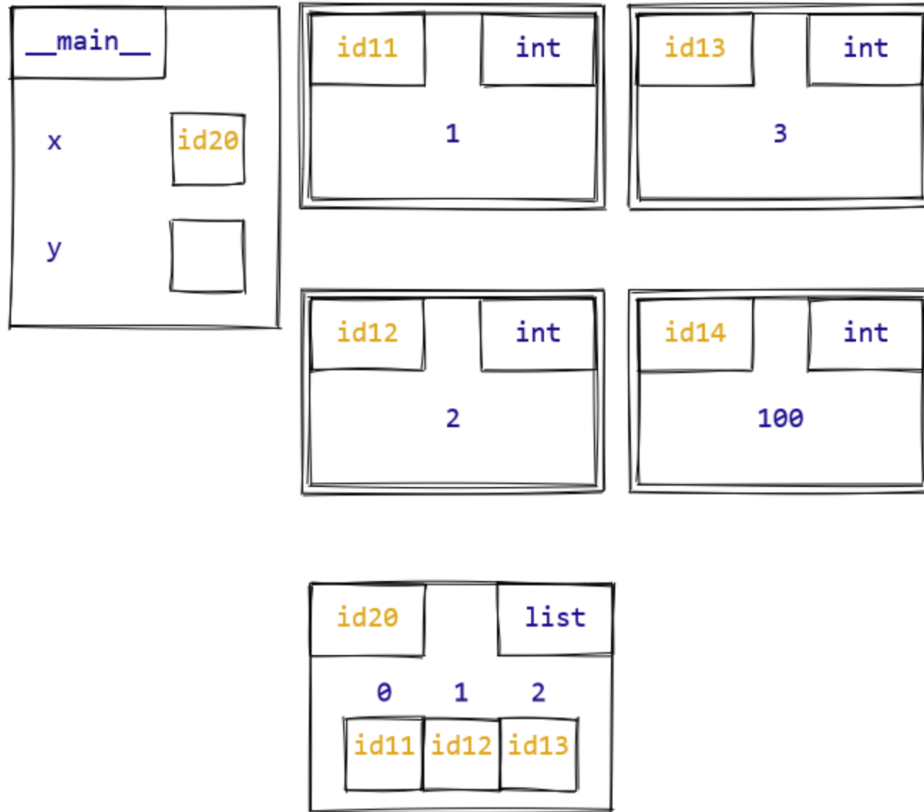


After the code above is executed, which of the following expressions evaluate to True? Circle those expression(s).

- (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`

2. Consider this code

```
1 x = {1, 2, 3}
2 y = x
3 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`



3. .

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



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

```
lst = [[1, 2, 88], [1]]
```

4. .

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



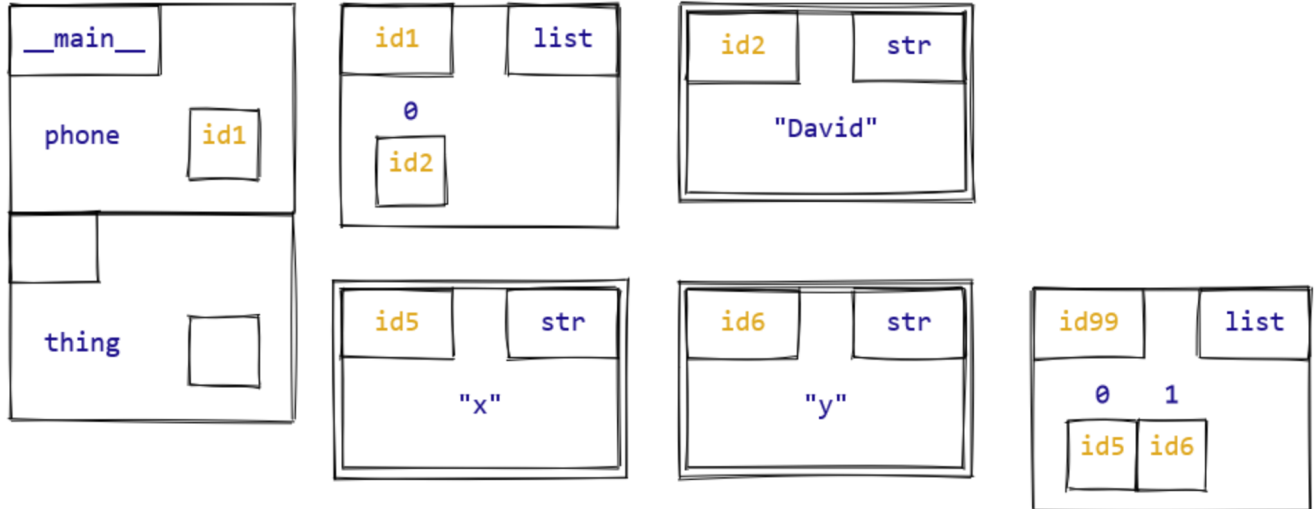
What is the value of `lst` at the end of this code snippet?

## 4 Exercise 4: Functions and the call stack

For each snippet of code below, draw the memory model diagram for the state of the program *immediately before* the function (`f1` or `f2`) returns. Then, write what would be displayed in the Python console.

1. .

```
1 def f1(thing: List[str]) -> None:
2     thing = ['x', 'y'] + thing
3
4
5 # In Python console
6 >>> phone = ['David']
7 >>> f1(phone)
8 >>> phone
```



1. .

```
1 def f2(thing: List[str]) -> None:
2     thing.extend(['x', 'y'])
3
4
5 # In Python console
6 >>> phone = ['David']
7 >>> f2(phone)
8 >>> phone
```

