

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 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.

Question 1
a = 5
b = 4
c = 6

__main__

a

id1

b

c

id1

int

5

id2

int

id3

int

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

__main__

a

id1

b

c

id1

int

5

id2

int

id3

list

012

Question 3
a = 5
b = [[a]]

__main__

a

id1

b

id1

int

5

id2

list

id3

list

id4

list

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.

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) behave correctly if `a` and `b` referred to mutable values like `lists` instead of `ints`?

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:

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.

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.)

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

__main__

x

id20

y

id11

int

1

id12

int

2

id13

int

3

id14

int

4

id20

list

012

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

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

__main__

x

id20

y

id11

int

1

id12

int

2

id13

int

3

id14

int

4

id20

set

{id11, id12, id13}

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`

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

__main__

lst

id10

item

id10

list

01

id11

list

01

id12

list

01

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?

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

__main__

lst

id10

item

id10

list

01

id11

list

01

id12

list

01

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?

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`/`f2`) returns. Then, write what would be displayed in the Python console.

def f1(thing: list[str]) -> None:
 thing = {'x', 'y'} + thing

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

__main__

phone

id1

thing

id1

list

0

id2

str

"David"

id5

str

"x"

id6

str

"y"

id99

list

01

def f2(thing: list[str]) -> None:
 thing.extend(['x', 'y'])

In Python console
>>> phone = ['David']
>>> f2(phone)
>>> phone

__main__

phone

id1

thing

id1

list

0

id2

str

"David"

id5

str

"x"

id6

str

"y"

id99

list

01