CS 120: Intro to Computer Programming II

In-Class Activity - 02 References - Day 2

Throughout today, keep quoting this rule back to yourself, over and over - it will help you complete the assignments!

Any time we have an assignment statement,

```
x = value
```

what that really means in Python is:

> **"Figure out what this `value` is; it will end up being a reference to an object. Take the name `x` (that is, the variable `x`) and fill it up with a reference, which will point at this object."**

Activity 1 - Turn in this one

Before anybody gives the "right" answer out, take a poll of everybody in the group. What do they think the answer to this quetsion is? After you've taken the poll, then discuss it more widely: what is the group consensus about the correct answer?

In the video, we considered the assignment statement

```
x = y
```

After this assignment statement, does `x` point to the variable `y`, or to something else? **Make sure to explain your answer!**

Solution:

```
No!  \texttt{x} points at the {\bf same object} as \texttt{y} points to,
but does {\bf not} point to \texttt{y} itself.  It is {\bf impossible}
(in Python) to point to a variable.
```

Activity 2 - Turn in this one

Make sure that everybody discusses, in the group, what they think this code snippet will print out - **before** anybody executes it. Then, in a second step, work as a group to build a reference diagram to show how the variables are related to each other. Only execute this code **after** you've done both of the previous steps. What does it print out?

Turn in three things: (1) information about the discussion; (2) the reference diagram that the group built; and (3) the actual value printed out.

```
y = 100
x = y
z = x
y = 444
print(z)
```

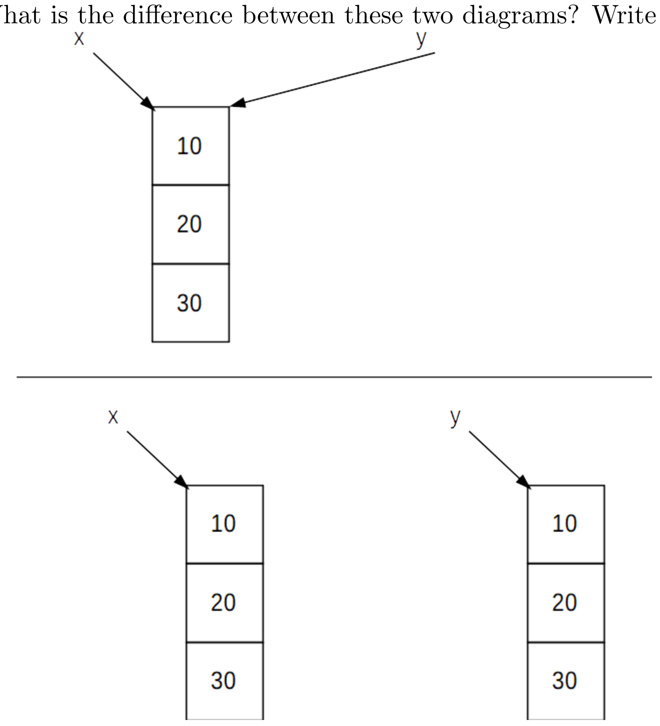
Solution: This will print out 100, not 444!

The reason for this is because the assignment statement copies references. Thus, in the first few lines of code, we create 3 different variables, all pointing at the same object (100). When we set `y` to 444, this changes **one** of the references, but `x,z` continue to point at 100.

(activity continues on the next page)

Activity 3 - Turn in this one

What is the difference between these two diagrams? Write some code which will create each one.



Solution: Upper diagram

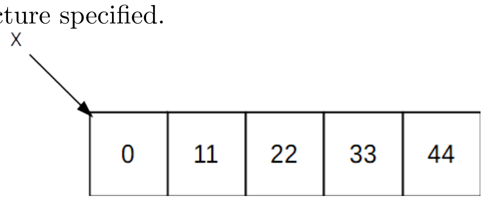
```
x = [10,20,30]
y = x
```

Lower diagram

```
x = [10,20,30]
y = [10,20,30]
```

Activity 4 - Turn in this one

The following diagram was built with a `for` loop. Fill in the missing lines, so that your code will build the picture specified.



```
____???____
for i in range(5):
    ____???____
```

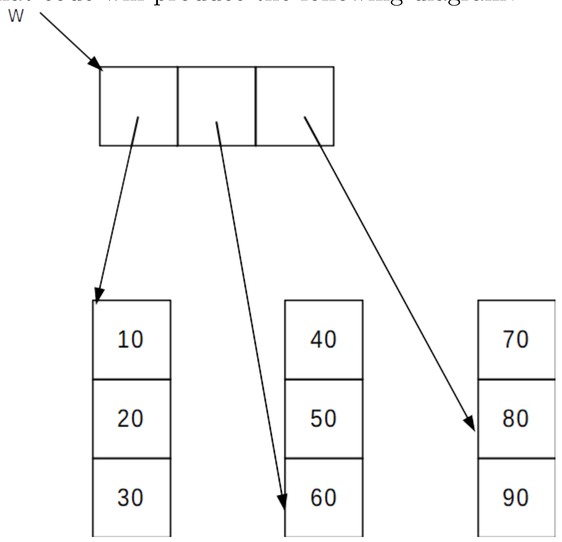
Solution:

```
x = []
for i in range(5):
    x.append(i*11)
```

(activity continues on the next page)

Activity 5 - Turn in this one

What code will produce the following diagram?



Solution:

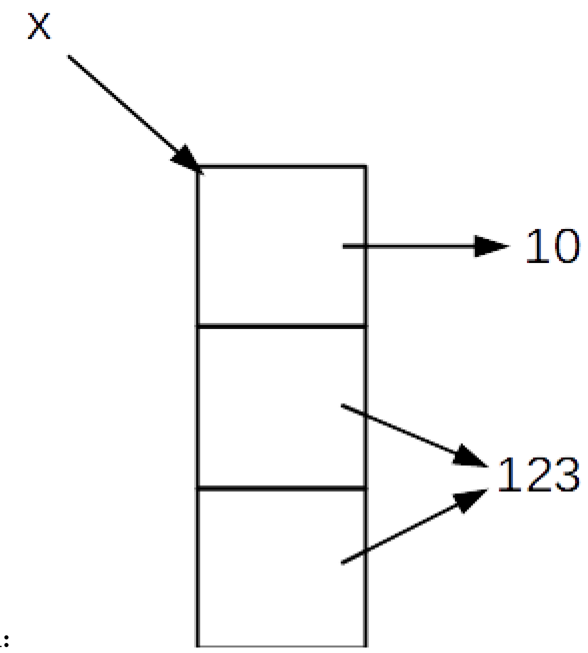
```
w = [ [10,20,30], [40,50,60], [70,80,90] ]
```

Challenge Activity - Do not turn in this one

Draw the data structure diagram for the following snippet of code. Execute the code carefully; remember what you know about how assignment works.

Hint: If you end up with a loop, then you've done something wrong!

```
x = [10,20,30]
x[2] = x[0]
x[0] = x[2]
x[2] = 123
x[1] = x[2]
```

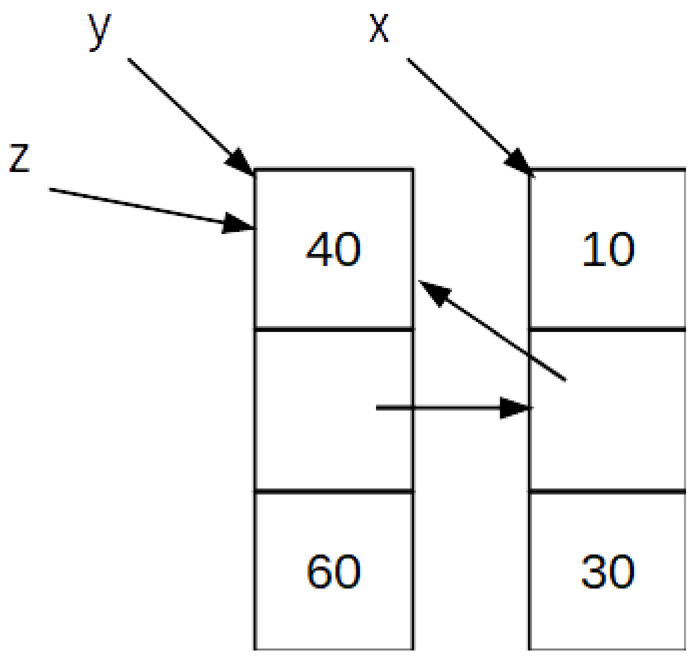


Solution:

Now, draw the data structure diagram for this second snippet of code. Again, simply execute the code according to the rules you've learned.

Hint: If you do **NOT** end up with a loop, then you've done something wrong!

```
x = [10,20,30]
y = [40, x,60]
z = y
z[1][2] = y
```



Solution: