Name:_____ **Partner:** _____

Python Activity 21: Scope

Variables have limited visibility inside and outside of functions.

Learning Objectives

Students will be able to:

Content:

- Define **scope** in python.
- Identify the scope of **local** and **global** variables.
- Predict how scope will impact variable assignment.

Process:

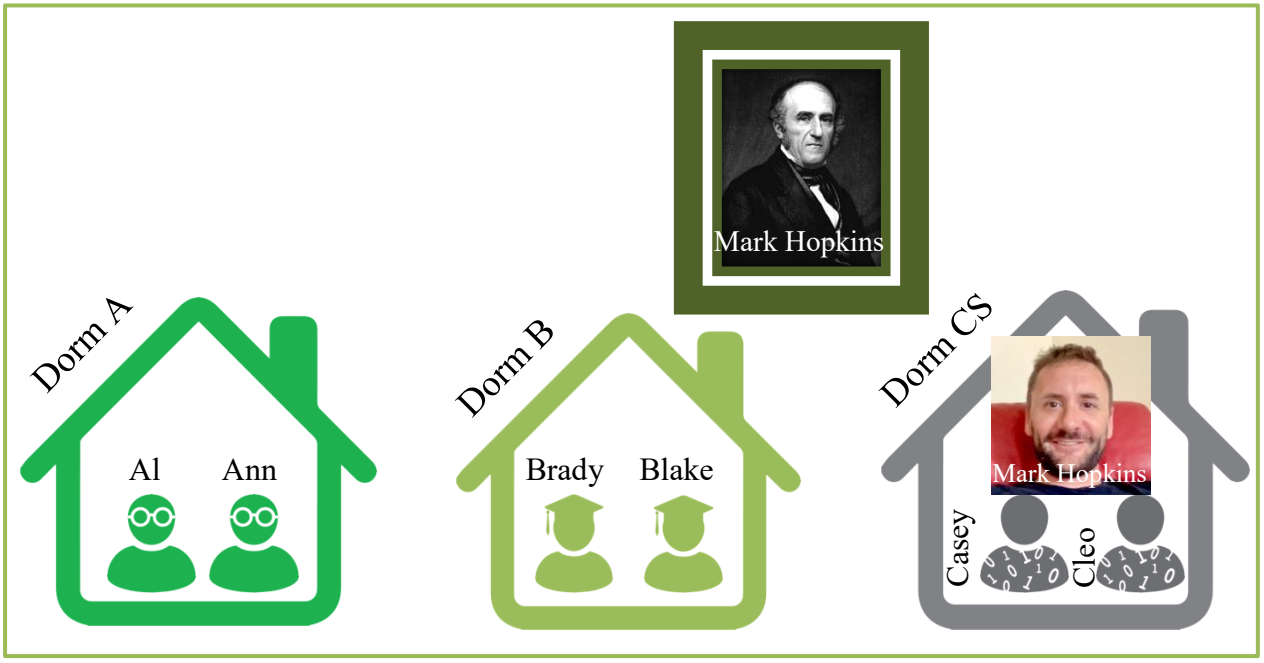
- Write code that properly assigns values to local and global variables.

Prior Knowledge

- Python concepts: assignment, functions, expressions

Concept Model:

Observe the following diagram, depicting three dorms. Two of which think of "Mark Hopkins" as referring to Mark Hopkins '1824, President of Williams College 1836-1872. The third dorm, Dorm CS, is full of Computer Science students who think "Mark Hopkins" refers to Professor Mark Hopkins who started working at Williams in 2022:



CM1. You overhear a conversation between 2 students, Ann and Cleo. Ann says, "Mark Hopkins was born in 1802." Cleo replies, "Mark Hopkins is a time traveler then!" Briefly explain why Cleo thinks this:

_____

_____

Examine the following python snippet that emulates the diagram above:

Code Example

```
mar_hop = 111119 # Mark Hopkins '1824 student ID number

def dorm_a():
     al = 223456 # Al's student ID number
     ann = 287654 # Ann's student ID number
     print(al, ann, mar_hop)
def dorm_b():
     brady = 277777 # Brady's student ID number
     blake = 288888 # Blake's student ID number
     print(brady, blake, mar_hop)
def dorm_cs():
     mar_hop = 998877 # Mark Hopkins '2022 student ID number
     casey = 212233 # Casey's student ID number
     cleo = 233444 # Cleo's student ID number
     print(casey, cleo, mar_hop)
```

CM2. If we were to call the function affiliated with Ann's dorm, `dorm_a()`, what do you expect would be printed?

_____

If we were to call the function affiliated with Cleo's dorm, `dorm_cs()`, what do you expect would be printed?

_____

How might the printed values for the variable `mar_hop` differ (do they?)? Why/not?

_____

_____

CM3. If we were to add the print statement `print(ann)` to the bottom of the `dorm_b()` and `dorm_cs()` functions, what do you predict will happen when we call these two functions?

_____

_____

FYI: The mapping of variable/function/object names to objects is limited in **scope**. Functions and classes all generate independent **frames** where these mappings are stored. This creates objects that can be seen in one frame, but not another.

Critical Thinking Questions:

1. Examine the following code below:

Code Example

```
# Question 1a
0 def triple(num):
1     multiplier = 3
2     return multiplier * num
3 answer = triple(5)
4 print(answer)
```

```
# Question 1b
0 multiplier = 3
1 def triple(num):
2     return multiplier * num
3 answer = triple(5)
4 print(answer)
```

a. Where does the assignment for the `multiplier` variable appear in the above code for Question 1a?

(i) **before** the function header

(ii) **in** the function body

(iii) **after** function body, before function call

(iv) **after** function body, after function call

What might the above code for Question 1a print to terminal? _____

b. Where does the assignment for the `multiplier` variable appear in the above code for Question 1b?

(i) **before** the function header

(ii) **in** the function body

(iii) **after** function body, before function call

(iv) **after** function body, after function call

What might the above code for Question 1b print to terminal? _____

Code Example

```
# Question 1c
0 def triple(num):
1     return multiplier * num
2 multiplier = 3
3 answer = triple(5)
4 print(answer)
```

```
# Question 1d
0 def triple(num):
1     return multiplier * num
2 answer = triple(5)
3 multiplier = 3
4 print(answer)
```

c. Where does the assignment for the `multiplier` variable appear in the above code for Question 1c?

(i) **before** the function header

(ii) **in** the function body

(iii) **after** function body, before function call

(iv) **after** function body, after function call

What might the above code for Question 1c print to terminal? _____

d. Where does the assignment for the `multiplier` variable appear in the above code for Question 1d?

(i) **before** the function header

(ii) **in** the function body

(iii) **after** function body, before function call

(iv) **after** function body, after function call

What might the above code for Question 1d print to terminal? _____

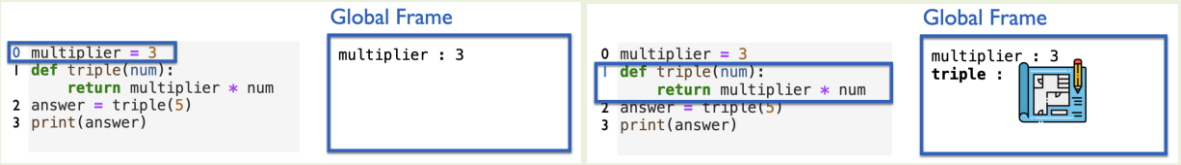
e. **Only one** of the above code examples results in a "`NameError: name 'multiplier' is not defined`" error. Which example *might* that be, and why?

_____

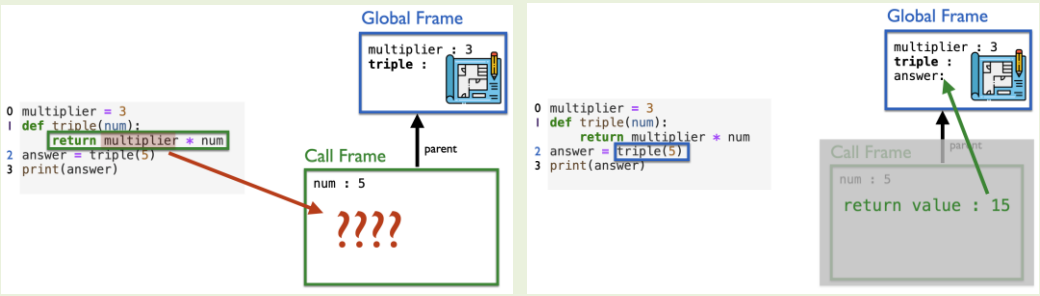
_____

Concept Model:
Observe your instructor describing how the **Function Frame Model** works. Below are provided a summary and a few snapshots of the illustrations:

By default, python reads code one line at a time, starting from line 0. At first, when variables are assigned, their values are stored in the **global frame**. Function `def`initions are treated like a single line of code. A `def` statement does not call the function, it just defines it. Effectively, it assigns the name of the function to a blueprint for computing the function.



To execute an assignment statement, python first computes the value of its right-hand side. When a function is called, a new frame is created to record the variables used by that function. First the values of the argument variables are recorded in the **call frame**. Then, the lines of the function are executed in order. To look up the value of a variable, first python looks in the call frame. If the variable isn't found in the call frame, then python looks in the **parent frame** (the frame we were in when the function was defined).



Ultimately, a **return value** is computed for the function call. The call frame is **destroyed** and the return value of the function call is assigned to the variable on the lefthand side of the assignment operator in the global frame.

Critical Thinking Questions:

2. Examine the following code below:

Code Example

```
0   def triple(num):
1       return multiplier * num
2   answer = triple(5)
3   multiplier = 3
4   print(answer)
```

a. What is recorded in the **global frame** after line 1 is initially seen by python? _____

b. What happens to the frames at line 2?

_____

c. What value is recorded for `multiplier` when `triple(..)` is called on line 2?

_____

d. What might happen when we run this code?

_____

3. Examine the following code below:

Code Example

```
multiplier = 3
def mystery(num):
    return multiplier * num
multiplier = 2
answer = mystery(5)
print(answer)
```

a. What is printed to the computer screen in the above example? _____

b. Why? _____

4. Examine the following code below:

Code Example

```
list = 2468
list_str = list("whodoweappreciate")
print(list, list_str)
```

a. What is printed to the computer screen in the above example? _____

b. Why? _____

5. Examine the following code below:

Code Example

```
a = 3
b = 4
def square(a):
    return a * a
c = square(a) + square(b)
c = pow(c, 0.5)
print(c)
```

```
a = 3
b = 4
def square(a):
    return a * b
c = square(a) + square(b)
c = pow(c, 0.5)
print(c)
```

a. What is printed to the computer screen in the left example? _____

b. Why? _____

c. How do the left and right examples differ? _____

d. How will these two changes impact the output displayed to the computer?

_____