#### Name:\_\_\_\_

# Partner:

## Python Activity 5: Value Returning Functions

Sometimes, we want to reuse code blocks that calculate or recombine values!

#### Learning Objectives

Students will be able to: *Content:* 

- Explain the meaning and purpose of a value returning function *Process:*
- Write code that includes function definitions and function calls

### **Prior Knowledge**

• None-returning functions

**FYI:** So far, the functions you have created print the results within the function. They do not send back any information to the original calling code. Functions that do not send back information are known as **None-returning functions**. Functions often send back or *return* a result and are known as **value returning functions**.

1. Carefully examine the code below, we'll run it as a class.

Python Program
<b>def</b> get_square(a):
square = a**2
return square
# Main code
my_square = get_square(5)
<pre>print("The square of 5 is " + str(my_square))</pre>

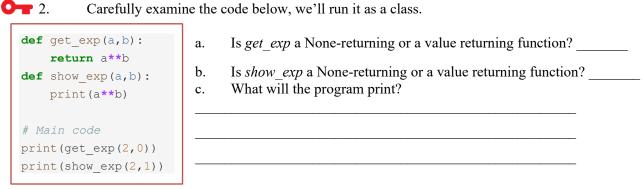
a. Circle the line of code from the program that includes the **function call** to *get\_square*.

**O** b. In a None-returning function, the function call is on a line by itself. Why is this function call placed on the right-hand-side of an assignment statement?

- c. What are the arguments used for the function call?
- d. What does the program do?

• Circle the keyword in the function that we didn't see in previous None-returning functions.

f. Is the function a **None-returning function** or a **value returning function**?



d. Execute and run the program. Does your prediction in (c) match the actual output? Why?

## Application Questions: Use the Python Interpreter to check your work

- 1. Carefully examine and then complete the following *Python* program:
  - The program prompts the user to enter their name.
  - It also asks the user for a number of times to print their name:
  - The program prints the user's name as many times as the user indicated.
  - Try writing this twice: using a None-returning and a value returning function!