

Name: \_\_\_\_\_ CM: \_\_\_\_\_ Section: \_\_\_\_\_ Grade: \_\_\_\_\_ of 10  
 (CM = Campus Mailbox #, like 2843. Section is 1, 2, 3, or 4 (Fisher, Mutchler, Fisher, Alangar) - see your schedule. Leave Grade blank.)

1. **Scope.** Consider the code shown to the right. The code will produce an error message when *main* runs. What error message will be produced? That is, what is wrong with the code?

```
def main():
    a = 5
    foo()

def foo():
    print(a)
```

*NameError: name 'a' is not defined*

That is, the name (variable) *a* in *foo* is undefined.

2. **Scope.** Consider the code shown below. The code will produce an error message when *main* runs. What error message will be produced? That is, what is wrong with the code?

```
def main():
    foo(4, 10)
    print(c)

def foo(a, b):
    c = a + b
```

*NameError: name 'c' is not defined*

The name (variable) *c* in *main* is undefined.

```
def main():
    a = 5
    b = 3
    foo(a, b)
    print(a)
    print(b)

def foo(b, a):
    print(a)
    print(b)
    a = 100
    b = 300
```

3. **Calling functions, arguments/parameters.** Consider the code shown to the right. What does it display on the console when *main* runs?

3  
5  
5  
3

4. **Calling functions, arguments/parameters, scope.** Consider the code shown to the right. Does the code produce an error message when *main* runs? If so, write *Error*. If not, show what gets displayed on the console.

10  
33

```
def main():
    a = 10
    x = 33
    foo(a, x)

def foo(x, c):
    print(x)
    print(c)
```

5. **Returned values.** The code shown immediately to the right is silly. Explain why.

Function *foo* returns a value but *main* pays no attention to the returned value.

```
def main():
    foo(7, 2)

def foo(x, y):
    return x + y

main()
```

6. **Returned values.** Consider the code shown to the right. What does it display on the console when *main* runs? Hint: TWO lines are displayed.

25  
None

```
def blah(x):
    print(x * x)

def main():
    print(blah(5))
```

7. **Using functions.** Suppose that your module contains the function shown to the right, and assume that it has been implemented correctly (per the specification in its doc-string).

```
def product_of_digits(x, y):
    """
    Returns (the sum of the digits of x)
    times (the sum of the digits of y).
    Example: If x is 12 and y is 501,
    this function returns 18.
    """
```

```
def sum_of_digits(number):
    """
    What comes in: An integer.
    What goes out: Returns the sum of the digits
    in the given integer.
    Example: If the integer is 83135, this function
    returns (8 + 3 + 1 + 3 + 5), which is 20.
    """
    <code hidden>
```

Implement the `product_of_digits` function defined to the left, per the specification in its doc-string. **Be sure that you understand the example in the doc-string!**

*Requirement:* You must call `sum_of_digits` to do your work!

*Hint:* **the code is a one-liner!**

**WRITE YOUR CODE BELOW HERE:**

```
return sum_of_digits(x) * sum_of_digits(y)
```

8. **Using functions.** Continuing the previous problem, now implement the function shown to the right. **Be sure that you understand the example in the doc-string!**

```
def sum_of_digits_of_product(x, y):
    """
    Returns the sum of the digits of
    (x times (the sum of the digits of y)).
    Example: If x is 12 and y is 501,
    this function returns 9.
    """
```

**WRITE YOUR CODE BELOW HERE:**

```
return sum_of_digits(x * sum_of_digits(y))
```

```
size = 10
for k in range(3):
    size = size + 5
    print(k, size)
    size = size - k
    print(size)
```

9. **Tracing code by hand.** You are starting to learn how to “think like a computer”, tracing code and keeping track of the state of the computer. Do that now with the code snippet shown to the left, by showing what gets displayed on the console when the code runs.

Note: I have put extra spaces in the output to make it easier to read.

```
0    15
15
1    20
19
2    24
22
```

10. The specification of a function tells which things? Mark all that apply.

**YES** Any side effects of the function

**NO** How the function works

**YES** What goes in

**YES** What comes out