Name:_____**SOLUTION**

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.

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

```
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
>
> 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.

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

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

> 10
>
> 33

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

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

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

main()
```

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

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.

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

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

> 25
>
> None

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

7. **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
```

8. 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*