

Name: _____ **SOLUTION** _____ Section: **1 2 3 4**

1 = Mutchler, 1st-2nd periods. **2** = Mutchler, 3rd-4th periods. **3** = Anderson, 7th-8th periods. **4** = Anderson, 9th-10th periods.

Use this practice to help you prepare for the Paper-and-Pencil portion of Exam 1. **Answer all questions.** Make additional notes as desired. **Not sure of an answer?** Ask your instructor to explain in class and revise as needed then.

Throughout, where you are asked to “circle your choice”, you can underline or circle it (whichever you prefer).

1. Consider the **secret** function defined to the right. What are the values of:

```
def secret(x):
    y = (x + 1) ** 2
    return y
```

- a. **secret(2)** _____ **9** _____
- b. **secret(secret(2))** _____ **100** _____

2. Consider the **mystery** function defined to the right. What are the values of:

```
def mystery(x, y):
    result = x + (3 * y)
    return result
```

- a. **mystery(5, 2)** _____ **11** _____
- b. **mystery(2, 5)** _____ **17** _____

3. Consider the code snippets defined below. They are contrived examples with poor style but will run. For each, what does it print when *main* runs? (Each is an independent problem.)

```
def main():
    x = 5
    foo(x)
    print(x)

def foo(x):
    print(x)
    return x ** 3
```

Prints: _____ **5** _____

_____ **5** _____

```
def main():
    x = 5
    y = foo(x)
    print(y)

def foo(x):
    x = 10
    print(x)
    return x ** 3
```

_____ **10** _____

_____ **1000** _____

```
def main():
    x = 5
    x = foo(x)
    print(x)

def foo(x):
    print(x)
    return x ** 3
```

_____ **5** _____

_____ **125** _____

8. What gets printed when *main* is called in the program shown to the right? (Pay close attention to the order in which the statements are executed. **Write the output in a column to the left of the program.**)

Output

2 3

2 3

2 3

```
def main():
    a = 2
    b = 3

    foo1()
    print(a, b)

    foo2(a, b)
    print(a, b)

    foo3(a, b)
    print(a, b)

def foo1():
    a = 88
    b = 99

def foo2(a, b):
    a = 400
    b = 500

def foo3(x, y):
    x = 44
    y = 55
```

9. True or False: As a **user** of a function (that is, as someone who will **call** the function), you *don't need to know how the function is implemented*; you just need to know the **specification** of the function. True False (circle your choice)
10. List **two** reasons why functions are useful and important.

They help organize the code, which makes it easier to get correct and maintain.

They allow for code re-use, by allowing the function to be called multiple times with different values for the parameters.