

Name: _____

Use this quiz to help make sure you understand the videos/reading. **Answer all questions.** Make additional notes as desired. **Not sure of an answer?** Ask your instructor to explain in class.

Video: **The Debugger** [13:04 minutes]

Note: Don't try to learn all the details of the Debugger – just breeze through this video to get the ideas, and return to it later as needed. These questions highlight the main points:

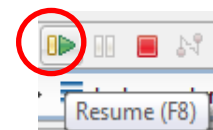
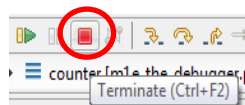
1. When the coder finds and fixes errors in her code, we call that: _____.
2. Technically, **testing** is the process one does **after** producing the code (or at least after producing its first version), while **debugging** is what the coder does to find and fix errors **while** producing the code.
However, we will use the terms **debugging** and **testing** interchangeably in CSSE 120. **Yes** **No** (circle your choice)
3. True or false: A debugger lets you set a **breakpoint** and run the program to that point, pausing the execution at the breakpoint. **True** **False** (circle your choice)
4. True or false: A debugger lets you **step** through a program, line by line. **True** **False** (circle your choice)
5. True or false: A debugger lets you **see the values of the variables** in your program and how they change as you step through program. **True** **False** (circle your choice)
6. Examine the code snippet to the right.

- a. At what lines have breakpoints been set?
- b. At what line is the program currently paused?
- c. At this point:
 - Has the program just executed line 37, or
 - Is the program just about to execute line 37?
 (Circle your choice.)

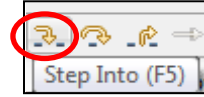
```

21 def debugger_example(n, color):
22     """
26     window = zg.GraphWin('A window with
27     x = 100
28     y = 50
29     radius = 25
30     for k in range(n):
31         sink = math.sin(k)
32         k_to_kth = k ** k
33         print(k, sink, k_to_kth, x, y)
34
35         p = zg.Point(x, y)
36         c = zg.Circle(p, radius)
37         c.setFill(color)
38         c.draw(window)
39
40         x = x + (2 * radius)
41         y = y + radius
42
  
```

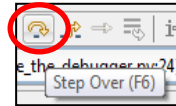
7. State, very briefly (just a few words for each is enough), what each of the following circled-in-red symbols do in the debugger:



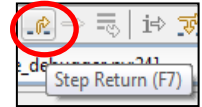
8. The **Step Into** button runs the next statement in the program.
Explain, very briefly:



- a. How is the **Step Over** button different (in what it does) from Step Into?



- b. How is the **Step Return** button different (in what it does) from Step Into?



Handout: **Functions with Parameters, and Namespaces**

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

```
def main():
    hello()
    goodbye()
    hello_and_goodbye()
    goodbye()

def hello():
    print('Hello!')

def goodbye():
    print('Ciao!')

def hello_and_goodbye():
    print('Here is stuff!')
    goodbye()
    hello()
    hello()
    print('Here is more!')
    hello()
    goodbye()
```

Output

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

```
def main():
    big()
    bigger()
    biggest()
    big()

def big():
    print('basketball')

def bigger():
    print('truck')
    big()

def biggest():
    print('house')
    bigger()
    big()
```

Output

11. For each of the following boxes:

- If the code is correct, state what gets printed when main runs.
- If the code is wrong, explain why.

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

Correct? If so, prints _____

Wrong?

If so, explain why:

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

Correct? If so, prints _____

Wrong?

If so, explain why:

```
def main():  
    x = foo('help')  
    print(x)  
  
def foo(m):  
    return m ** 3
```

Correct? If so, prints _____

Wrong?

If so, explain why:

12. The code in the box to the right has syntax errors: it causes big red **X** error message(s). Circle the line(s) that will have red **X** error message(s) beside them and explain why those line(s) will have those **Xs**.

```
def main():  
    foo()  
    print(n)  
    print(m)  
  
def foo():  
    n = 3  
    m = 1  
    return m
```

13. Suppose you want to write a function called *foo* that has two *zg.Point* objects sent to it and does something with them. Write the “header” line of *foo*, that is the line that begins with **def**.

def _____:

14. Suppose you want to write a function called *blah* that takes a *zg.Point* object and a *zg.GraphWin* object (in that order) and does something with them. Write the “header” line of *blah*, that is the line that begins with **def**.

def _____:

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

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

    m = do_it(a, b)
    print(m)

    m = do_it(b, a)
    print(m)

    m = do_it(a, a)
    print(m)

    m = do_it(b, b)
    print(m)

    c = do_it(b, a)
    m = do_it(c, a)
    print(m)

    b = do_it(b, a)
    m = do_it(b, a)
    print(m)

def do_it(x, y):
    return x ** y
```

Output

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

```
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
```

Output

Video: *The Accumulator Pattern – Summing* [13:04 minutes]

17. Trace the following snippet of code **by hand** (no fair typing it into a program), and **show what gets printed**:

```
total = 0
for k in range(5):
    total = total + (k + 10)
    print(k, total)

print('The sum 10 + 11 + 12 + 13 + 14 is')
print(total)
```

What gets printed?

The sum 10 + 11 + 12 + 13 + 14 is

18. Write a snippet of code that calculates:

sine(3) + sine(4) + sine(5) + ... + sine(500)