Brian Stumbaugh

HW chpt. 3

COSC 1336 sec 006

1. How do functions help you to reuse code in program?

When an operation is used in several places throughout a program, a function can be written to perform that operation, and can be executed any time it is needed. This saves you from having to write the same code multiple times throughout the program.

3. When a function is executing, what happens when the end of the function block is reached?

When the end is reached, the interpreter jumps back to the part of the program that called the function, and the program continues executing where it had left off.

6. Why do global variables make a program difficult to debug?

Any statement in a program file can change the value of a global variable.

Algorithm Workbench 2 and 4

2. Examine the following function header, and then write a statement that calls the function, passing 12 as an argument.

def main():

value = 12

show\_value(value) # I think this was the original question

def show\_value(quantity):

result = quantity

print(result)

main() #this was wrong. answer is : show\_value(12)

4. What will the following program display?

def main() :

x = 1

y = 3.4

print(x, y)

change\_us(x, y)

print(x, y)

def change\_us(a, b):

a = 0

b = 0

print(a, b)

main()

It will show: 1 3.4

0 0

1 3.4

The main function creates two variables and assigns them values (x = 1 , y = 3.4), and then prints them showing 1 3.4

Then the (x, y) variables are passed as an argument to the change\_us function, and are now assigned as (a,b). The change\_me function then takes (a,b) and assigns them both new values of 0.

It prints 0 0.

Then control of the program is returned to the main function, where it finishes by executing the unchanged variables of (x, y) as 1 3.4

#Id love some feedback on my terminology as well. Perhaps I could just ask you about it after class tom