Metropolitan State University

ICS 140 Computational Thinking with Programming

Class Exercise 8

**Lecture Section**

1. What is a function?

Section of code that performs a specific action that can be called when needed

1. What are the advantages of using functions (list at least 2)?

Better teamwork

Makes code more readable

Don’t have to write repeat code to run the same thing

1. What is the key word used to define a function?

def

1. How do you format your code to indicate what is contained in a function?

Using tabs (indentation) to keep it in the same block

1. What is a local variable?

Variable that is created withing and can be used inside the function but does not exist outside the function

1. How do you define your function to accept arguments?

Put a variable inside the parenthesis

1. How do you define the function’s output?

Use a return statement

**Function Invocations**

For the following questions, write the expected output:

1. What would the following code print?

def f1():

    print('1')

    print('2')

def f2():

    f1()

    print('3')

def f3():

    f2()

    f2()

    f1()

f3()

1

2

3

1

2

3

1

2

1. Use the following function to evaluate what the program would do for the following function calls:

def f4(a):

    print(a \* a)

1. f4(4) - 16
2. f4() – Would not run properly
3. f4(10, 20) - 120
4. The following function definitions or calling code have mistakes. Identify what is wrong with each one:
   1. define f():  
       a = 1 **define is not a proper keyword**
   2. Def f():  
       B = 2 **capital D will not work, keywords are case sensitive**
   3. def g()  
       C = 3 **Forgot a colon**
   4. def f(a, b, c):  
       print(str(a) + “/”+ str(b) = “/” + str(c))  
      f(1,2) **incorrect number of inputs when calling the function with f(1,2)**
   5. def f(a; b; c):  
       print(a, b, c) **Arguments need to be separated by a comma**
   6. def f(a b):  
       print(a, b)  **Arguments need to be separated by a comma**

**Writing Functions**

For the following programs, write a function to meet the requirements. If these concepts do not make sense yet, check the lab for this week. The lab works through each component of building and using functions in Python.

1. Define a function called **greeting()**. It should accept a parameter called **name** and print “hello, <**name**>” whenever it is called.

Def greeting(name):

Print(“hello”, name)

1. Define a function called **full\_name()**. It should accept parameters of **first\_name** and **last\_name**. It should return a concatenated string combining the 2 inputs with a space between them.

Def full\_name(first\_name, last\_name):

Print(first\_name, “ “, last\_name)

1. Define a function called **max()**. It should accept 2 integer parameters and return the larger of the 2.
2. def max(a,b):
3. if a > b:
4. return a
5. else:
6. return b
7. number1 = int(input("Enter a number: "))
8. number2 = int(input("Enter another number: "))
9. print(max(number1, number2))
10. Define a function called **odd\_even()**. It should accept an integer as a parameter and return a string of either “even” or “odd” based on the provided number.
11. def odd\_even(num):
12. remainder = num % 2
13. if remainder == 0:
14. print("Even")
15. else:
16. print("Odd")
17. odd\_even(int(input("please enter a number: ")))
18. Define a function called **currency()**. It should accept a float as a parameter and return a formatted string starting with a dollar sign and 2 decimal points.

def currency(num):

    print("Your number is", "${:.2f}".format(num))

    return num

input = float(input("Please enter a number: "))

currency(input)