Metropolitan State University

ICS 140 Computational Thinking with Programming

Class Exercise 8

**Lecture Section**

1. What is a function?

A reusable piece of code that performs a specific task.

1. What are the advantages of using functions (list at least 2)?
2. Code reusability, modular code
3. 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?

indentation

1. What is a local variable?

Variables defined within a specific function

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

Adding variables at the end of defining a function

EX: def funtion1(variable):

Blah blah

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

Return function

**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

1. f4()

type error

1. f4(10, 20)

type error

1. The following function definitions or calling code have mistakes. Identify what is wrong with each one:
   1. define f():  
       a = 1

define is wrong for defining a function should be def

* 1. Def f():  
      B = 2

Capital D in def

* 1. def g()  
      C = 3

no parameter in function g()

* 1. def f(a, b, c):  
      print(str(a) + “/”+ str(b) = “/” + str(c))  
     f(1,2)

not calling the function with all parameters

* 1. def f(a; b; c):  
      print(a, b, c)

should be commas not semi colans

* 1. def f(a b):  
      print(a, b)

no comma between a and b when defining function parameters

**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):

Return first\_name +’ ‘ + last\_name

1. Define a function called **max()**. It should accept 2 integer parameters and return the larger of the 2.

Def max(a: int,b: int):

if a > b:

return a

else:

return b

1. 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.

Def Odd\_even(number):

If number % 2 == 0:

Return ‘Even’

Else:

Return ‘odd’

1. 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(money: float) -> str:

Return ‘${:.2f}’.format(money)