1 Define a function that returns the value of up to a 4th order polynomial for a given input and set of parameters for the polynomial. Make the linear terms required inputs to the function; make the other terms optional inputs.

def polynomial(x,a,b,c=0,d=0,e=0):

"""Returns the value of up to a 4th order polynomial for a given input and set of

parameters for the polynomial"""

try:f=a+(b\*x)+(c\*x\*\*2)+(d\*x\*\*3)+(e\*x\*\*4)

except:print("Numbers required")

return f

print(polynomial(1,1,1))

Output: 2

2 Define a function that returns a product of an unknown set of numbers.

def prods(\*data):

from numpy import prod

a=[\*data]

b=prod(a)

print(b)

return

data=[2,3,6,8,10]

prods(\*data)

Output: 2880

3 (a) What does the ‘\*’ operator do as a prefix?

The first single special argument when the function takes a variable number of arguments when defining it; calls the function during the function call. The first function may have any name but must be prefixed by an asterix

3 (b) What does the ‘\*\*’ operator do as a prefix?

When using a variable number of unknown keyword arguments, otherwise the same as the \*.

3 (c) What is the difference between args and kwargs?

Args pass a non-keyworded, variable-length argument list to a function, and kwargs pass a keyworded, variable-length argument list.

3 (d) Why might you use kwargs instead of args? Give an example.

Kwargs are useful if you don’t know the names of parameters in advance.

4 (a) What properties do first class objects have in Python?

They are handled uniformly throughout. They can be stored in a variable, passed as a parameter to a function, and, if the object is a function it can be applied to other functions.

4 (b) In your own words, describe the meaning of global scope, local scope, and module scope.

Scope required a function of some sort. Global scope is all variables outside of the function. Local scope are variables defined inside of the function, and those variable definitions will disappear when the function disappears. Module scope is related to global scope, because all global variables are global only to the module (.py file) where they exist, not global to Python overall.

4 (c) What is the value of ‘var’ at each print statement in this code snippet?

var = 12

print var var = 12

def func(var):

print var var = 12

var = 6

print var var = 6

def funct (Var)L

print var var = 12

var = 9

print var var = 9

(funct(var)

funct(var)

print var var = 6

5 What is recursion in Python? Why does Python implement default limits for recursion?

Recursion is when a function calls itself. The default limit for a recursion in Python is 1,000, although the user can modify the recursion length with the sys.recursionlimit() tool. This prevents the function from executing infinitely, which would break the code and system memory. Fixed points in a recursion, which are defined such that x is a fixed point of f if and only if x == f(x), are important for grounding recursion and ensuring it does not compute too many times; otherwise the recursion limit will kick in.

6 Write a lambda function named sum to calculate the sum of a + b

sum = (lambda a, b: a + b)(1,2)

print(sum)

3

7 In your own words describe generators. Thinking of topics covered in this class, when might this be

useful?

A generator returns an answer for a given value but withholds executing it until prompted to do so. Generators can be used to calculate large chunks of data but only display small chunks of it. For example, if I wanted the thousandth term in a list of answers (Say, distinct probabilities of particle interaction), I could use a generator to yield that specific result rather than the entire list.

8 What is one concept you found difficult in the reading?

I do not understand the purpose of lambda functions at all.