1. What will be the output of the following piece of code?

d = { 1:0, 2:1, 3:2, 0:1 }

x = 0

for y in range(len(d)):

x = d[x]

print(x)

The output of the python script will be ‘2101’

1. What will be the output of the following piece of code?

v = 1 + 1 // 2 + 1 / 2 + 2

print(v)

the calculation will result in 4.5

1. What will be the output of the following code?

t = (2,)

t = t[0] + t[0]

print(t)

the output will be 4 as the value of t will be 4 by end of the calculation

1. fix the error(s) in the following piece of code

x = 16

while x > 0:

print(‘\*’,end=?)

x //= 2

this is as about as I could get it to the result wanted

x = 6

while x > 0:

print('\*' \* x)

x //= 2

1. What will be the output of the following snippet?

d = { ‘one’:1, ‘three’:3, ‘two’:2 }

for k in sorted(d.values()):

print(k,end=’ ‘)

the out put would be ‘1 2 3’ or the answer a

6 You have a function heading: def fun(a,b,c=0). Which of the following is a proper

call to the function?

the answter would be d. fun(a=1,b=0,c=0)

7.Fill in the blanks so the output is 5

\_\_\_\_\_\_\_\_ newclass:

newclass = 1

def \_\_init\_\_(self,v = 2):

self.v = v + newclass.newclass

\_\_\_\_\_\_\_\_\_.newclass += 1

def set(self,v):

self.v += v

newclass.newclass += 1

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

a = newclass()

a.set(2)

print(a.v)

here is the code with the blanks filled in.

class newclass:

newclass = 1

def \_\_init\_\_(self, v=2):

self.v = v + newclass.newclass

newclass.newclass += 1

def set(self, v):

self.v += v

newclass.newclass += 1

a = newclass()

a.set(2)

print(a.v)

8.

Finally always executes

Tuple a list that can’t change

Pass allows processing without a break or continue

Yield returns function’s values

Filestream uses a reading or writing marker to keep track of data processing

Dictonary data processed with key-value pair

9.

cookies = ('Chocolate Chip', 'Oatmeal Raisin', 'Sugar', 'Snickerdoodle', 'Peanut Butter')

prices = (1.50, 1.75, 1.25, 1.50, 1.75)

print("Our cookie selection:")

for i in range(len(cookies)):

    print(f"{i+1}. {cookies[i]}: ${prices[i]:.2f}")

new\_cookie = input("Would you like to add a new cookie to the menu? Enter the name of the cookie or 'no' to quit: ")

if new\_cookie.lower() != "no":

    new\_price = float(input(f"What is the price of {new\_cookie} (enter as a decimal)? "))

    cookies += (new\_cookie,)

    prices += (new\_price,)

    print(f"Thanks for adding {new\_cookie} to our menu! Here is your receipt:")

    print(f"{new\_cookie}: ${new\_price:.2f}")

    print("Updated menu:")

    for i in range(len(cookies)):

        print(f"{i+1}. {cookies[i]}: ${prices[i]:.2f}")

else:

    print("Thank you have a good day")

9.5

class Inventory:

    def \_\_init\_\_(self, item, price):

        self.\_\_item = item

        self.\_\_price = price

    def get\_item(self):

        return self.\_\_item

    def get\_price(self):

        return self.\_\_price

class Order:

    def \_\_init\_\_(self):

        self.\_\_items = []

    def add\_item(self, item):

        self.\_\_items.append(item)

    def get\_items(self):

        return self.\_\_items

    def grand\_total(self):

        subtotal = sum(item.get\_price() for item in self.\_\_items)

        tax = subtotal \* 0.05

        return subtotal + tax

chocolate\_chip = Inventory("Chocolate Chip", 1.50)

oatmeal\_raisin = Inventory("Oatmeal Raisin", 1.75)

sugar = Inventory("Sugar", 1.25)

snickerdoodle = Inventory("Snickerdoodle", 1.50)

peanut\_butter = Inventory("Peanut Butter", 1.75)

order = Order()

print("Welcome to the Rose State Café!")

while True:

    print("What would you like to order? Enter 'done' when you are finished.")

    item\_name = input("Item name: ")

    if item\_name.lower() == "done":

        break

    item\_price = float(input("Price: "))

    item = Inventory(item\_name, item\_price)

    order.add\_item(item)

grand\_total = order.grand\_total()

print(f"Your grand total with tax is ${grand\_total:.2f}. Thank you for your order!")