1. The meaning of the keyword argument is determined by:

* its position within the argument list
* its value
* its connection with existing variables
* the argument’s name specified along with its value

1. Which of the following sentences is true?

str1 = ‘string’

str2 = str1[:]

* str1 and str2 are different (but equal) strings
* str2 is longer than str1
* “>str1 and str2 are different names of the same string
* str1 is longer than str2

1. An operator able to check whether two values are equal, is coded as:

* =
* ==
* ===
* Is

1. The following snippet:

def f(par2, par1):

return par2 + par1

print(f(par2=1,2))

* will output 2
* will output 3
* will output 1
* is erroneous

1. What value will be assigned to the x variable?

z = 2

y = 1

x = y < z or z > y and y > z or z < y

* 0
* True
* 1
* False

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

str = ‘abcdef’

def fun(s):

del s[2]

return s

print(fun(str))

* abcef
* the program will cause an error
* abdef
* Acdef

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

x, y, z = 3, 2, 1

z, y, x = x, y, z

print(x,y,z)

* 2 1 3
* 1 2 3
* 1 2 2
* 3 2 1

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

a = True

b = False

a = a or b

b = a and b

a = a or b

print(a,b)

* True False
* True True
* False False
* False True

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

def fun(x):

return 1 if x % 2 != 0 else 2

print(fun(fun(1)))

* 2
* the code will cause a runtime error
* 1
* None

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

print(len((1,)))

* 0
* the code is erroneous
* 2
* 1

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 code will cause a runtime error
* 2
* 0
* 1

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

y=input()

x=input()

print(x+y)

if the user enters two lines containing 1 and 2 respectively?

* 21
* 12
* 2
* 3

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

print(“a”,”b”,”c”,sep=”‘”)

* a’b’c
* abc
* a b c
* the code is erroneous

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

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

print(v)

* 4.0
* 3.5
* 3
* 4

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

t = (1,)

t = t[0] + t[0]

print(t)

* (1,)
* 1
* (1, 1)
* 2

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

x = 16

while x > 0:

print(‘\*’,end=”)

x //= 2

* \*\*\*\*\*
* \*\*\*
* \*
* the code will enter an infinite loop

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=’ ‘)

* 1 2 3
* 3 2 1
* 2 3 1
* 3 2 1

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

print(len([i for i in range(0,-2)]))

* 0
* 2
* 3
* 1

1. Which of the following lines properly invokes the function defined as:

def fun(a,b,c=0)?

* fun(0):
* fun(b=0,b=0):
* fun(1,c=2):
* fun(a=1,b=0,c=0):

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

l = [1,2,3,4]

l = list(map(lambda x: 2\*x,l))

print(l)

* 10
* the snippet will cause a runtime error
* 1 2 3 4
* 2 4 6 8

1. How many stars will the following snippet send to the console?

i = 4

while i > 0 :

i -= 2

print(“\*”)

if i == 2:

break

else:

print(“\*”)

* 2
* 0
* 1
* the snippet will enter an infinite loop

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

t = (1, 2, 3, 4)

t = t[-2:-1]

t = t[-1]

print(t)

* 33
* (3)
* 3
* (3,)

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

d = {}

d[‘2’] = [1,2]

d[‘1’] = [3,4]

for x in d.keys():

print(d[x][1],end=””)

* 24
* 13
* 42
* 31

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

def fun(d,k,v):

d[k]=v

dc = {}

print(fun(dc,’1′,’v’))

* None
* 1
* the snippet is erroneous
* V

1. How many empty lines will the following snippet send to the console?

l = [[c for c in range(r)] for r in range(3)]

for x in l:

if len(x) < 2:

print()

* 1
* 0
* 2
* 3

1. Knowing that the function named m() resides in the module named f, and the code contains the following import statement, choose the right way to invoke the function:

from m import f

* the import statement is invalid
* mod.fun()
* mod:fun()
* fun()

1. The package directory/folder may contain a file intended to initialize the package. Its name is:

* \_\_init\_\_.py
* init.py
* \_\_init.py\_\_
* \_\_init\_\_.

1. The folder created by Python used to store pyc files is named:

* \_\_pycfiles\_\_
* \_\_pyc\_\_
* \_\_pycache\_\_
* \_\_cache\_\_

1. What will be the output of the following code, located in the file module.py?

print(\_\_name\_\_)

* main
* \_\_module.py\_\_
* module.py
* \_\_main\_\_

1. If you want to tell your module’s users that a particular variable should not be accessed directly, you may:

* start its name with a capital letter
* use its number instead of its name
* start its name with \_ or \_\_
* build its name with lowercase letters only

1. If there is a finally: branch inside the try: block, we can say that:

* it won’t be executed if no exception is raised
* it will always be executed
* branches is executed
* it will be executed when there is no else: branch

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

try:

raise Exception

except BaseException:

print(“a”,end=”)

else:

print(“b”,end=”)

finally:

print(“c”)

* a
* ab
* bc
* ac

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

class A:

def \_\_init\_\_(self,name):

self.name = name

a = A(“class”)

print(a)

* a number
* a string ending with a long hexadecimal number
* class
* Name

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

try:

raise Exception

except:

print(“c”)

except BaseException:

print(“a”)

except Exception:

print(“b”)

* it will an cause error
* b
* c
* a

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

class X:

pass

class Y(X):

pass

class Z(Y):

pass

x = X()

z = Z()

print(isinstance(x,Z),isinstance(z,X))

* False False
* True True
* True False
* False True

1. The following code prints:

x = “\”

print(len(x))

* 1
* the code will cause an error
* 2
* 3

1. The following code prints:

x = “””

“””

print(len(x))

* 2
* 1
* the code will cause an error
* 3

1. If the class constructor is declared as below, which one of the assignments is valid?

class Class:

def \_\_init\_\_(self):

pass

* object = Class(None)
* object = Class(1)
* object = Class(1,2)
* object = Class()

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

class A:

A = 1

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

self.v = v + A.A

A.A += 1

def set(self,v):

self.v += v

A.A += 1

return

a = A()

a.set(2)

print(a.v)

* 7
* 5
* 1
* 3

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

class A:

A = 1

def \_\_init\_\_(self):

self.a = 0

print(hasattr(A,’A’))

* True
* 0
* 1
* False

1. What will be the result of executing the following code?

class A:

pass

class B:

pass

class C(A,B):

pass

print(issubclass(C,A) and issubclass(C,B))

* it will print True
* it will raise an exception
* it will print an empty line
* it will print False

1. The sys.stdout stream is normally associated with:

* the screen
* a null device
* the keyboard
* the printer

1. What will be the effect of running the following code?

class A:

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

self.\_a = v + 1

a = A(0)

print(a.\_a)

* it will print 0
* it will print 1
* it will print 2
* it will raise an AttributeError exception

1. What will be the result of executing the following code?

class A:

def \_\_init\_\_(self):

Pass

def f(self):

return 1

def g():

return self.f()

a = A()

print(a.g())

* it will print 0
* it will print True
* it will print 1
* it will raise an exception

1. What will be the result of executing the following code?

class A:

def a(self):

print(‘a’)

class B:

def a(self):

print(‘b’)

class C(A,B):

def c(self):

self.a()

o = C()

o.c()

* it will print b
* it will print a
* it will raise an exception
* it will print c

1. The Exception class contains a property named args, and it is a:

* string
* tuple
* list
* Dictionary

1. What will be the result of executing the following code?

def I(n):

s = ”

for i in range(n):

s += ‘\*’

yield s

for x in I(3):

print(x,end=”)

* it will print \*\*\*
* it will print \*\*\*\*
* it will print \*
* it will print \*\*\*\*\*\*

1. What will be the result of executing the following code?

def a(x):

def b():

return x + x

return b

x = a(‘x’)

y = a(”)

print(x() + y())

* it will print xxxxxx
* it will print x
* it will print xx
* it will print xxxx

1. If s is a stream opened in read mode, the following line

q = s.readlines()

will assign q as a:

* string
* dictionary
* list
* Tuple

1. If you want to write a byte array’s content to a stream, you’d use:

* the write() method
* writebytearray() method
* the writefrom() method
* writeto() method