1. The simplest possible class definition in Python can be expressed as:
2. Class X:

Return

1. Class X:
2. Class X:

Pass

1. Class X: {}
2. Any class method must have at least:
3. There is no requirement as to the number of parameters that a class must have
4. Two parameters
5. Three parameters
6. One parameter
7. What is the expected output of the following code?

Lst = [x for x in range(4) ]

Map(lambda x: x+x,lst)

Print(lst)

1. [0,1,2,3]
2. The code will cause a runtime exception
3. [0,2,4,6]
4. [0,2,4,9]
5. What is the result of the following comparison?

‘12’>=12

1. The comparison causes a runtime exception/error
2. True
3. False
4. The result is undefined
5. Assuming that lst is a four-element list, is there any difference between these two statements?

del lst # the first line

del lst [ : ] # the second line

1. Yes, there is the first line deletes the list as a whole; the second line just empties the list
2. No, there is no difference
3. Yes, there is; the first line empties the list, the second line deletes the list as a whole
4. Yes, there is; the first line deletes the list as a whole, the second line removes all the elements except the first one
5. What can you deduce from the line below? (select two answers)

X = a.b.c.f()

1. The line is incorrect
2. The function being invoked is called a.b.c.f()
3. F() is located in subpackage of c of subpackage b of package a
4. Import a.b.c should be placed before that line
5. A compiler is a program designed to ( select two answers)
6. Translate the source code into machine code
7. Check the source code in order to see if it’s correct
8. Execute the source code
9. Rearrange the source code to make it clearer
10. The newly defined exception MyZeroDivisionError as seen below:

Class MyZeroDivisionError(ZeroDivisionError):

Pass

Is:

1. Completely independent from ZeroDivisionError (in the sense of class inheritance)
2. Exactly the same as ZeroDivisionError
3. More specific than ZeroDivisionError
4. More general than ZeroDivisionError
5. What will be the value of the i variable when the while loop finishes its execution?

i=0

while i !=0:

i = i -1

else:

i = i +1

1. 1
2. The variable becomes unavailable
3. 0
4. 2
5. What is the expected behavior of the following snippet?

def x(): # line 01

return 2 # line 02

x = 1 + x() # line 03

print(x) # line 04

It will

1. Cause a runtime exception on line 02
2. Cause a runtime exception on line 03
3. Cause a runtime exception on line 01
4. Print 3
5. Class A:

Def a(self):

Print(“A”, end =’’)

Def b(self):

Self.a()

Class B(A):

Def a(self):

Print(“B”, end=’’)

Def do(self):

Self.b()

Class C(A):

Def a(self):

Print(“C”, end=’’)

Def do(self):

Self.b()

B().do()

C().do()

1. CC
2. BC
3. AA
4. BB
5. If you want to obtain a list that is filled with the first five cubed natural numbers, which clause would you use?
6. [ x \*\* 3 for x in range(5) ]
7. [ for x \*\* 3 in range(5) ]
8. [ x = i\* i\* i for i in range(5) ]
9. [ x \*\* 3 for x in 1..5 ]
10. What is the expected behavior of the following code?

Def f(a):

For i in reversed(a):

Yield i + 1

For i in f([x for x in range(2)]):

Print(i, end=’’)

It will

1. Print 1 2
2. Print 2 1
3. Print <generator object f at (some hex digits) >
4. Cause a runtime exception
5. If you want to iterate through all of the dictionary’s keys which clause would you use instead of XXX?

for key in XXX:

print(key)

1. keys(dict)
2. dict.keys()
3. dict.keys
4. keys.dict
5. If a file is opened in the append mode, this means that ( select two answers )
6. The file doesn’t need to exist before opening it
7. If the file does not exist, it will be created
8. If the file exists, it will be written from its beginning
9. The stream associated with the file can be read
10. If any of a class’s components has a name that starts with two underscores(\_\_) then:
11. The class component has to be a class variable
12. The class component has to be an instance variable
13. The class component’s name will be mangled
14. The class component has to be a method
15. An entity initialized in the following way

Ent = 1,2,4,8

Is a:

1. List
2. Dictionary
3. Tuple
4. Regular variable
5. The following instruction:

(select two answers)

Assert expression

1. Is invalid
2. Raises AssertionError if the expression evaluates to False
3. Can be used anywhere
4. Stops the program immediately
5. If the following snippet is executed and the exception is raised as seen below:

Try:

# some code

raise Exception(“Error message”)

# some code

Except Exception as e:

Print(e)

Print(e.\_\_str\_\_())

You will see:

1. Two different non-empty lines
2. Two empty lines
3. Two identical non-empty lines
4. Four lines containing a system error message
5. What is the expected output of the following snippet?

Data = [1,2,3,4,5,6 ]

Idx = data.index(data[0] +3)

Print(idx)

1. The snippet will cause a runtime exception
2. The snippet will print 3
3. The snippet will print [3]
4. The snippet contains a syntax error
5. What is the expected output of the following snippet?

S = ‘abc’

For i in len(s):

S[i] = s[i].upper()

Print(s)

1. 1 2 3
2. The code will cause a runtime exception
3. abc
4. ABC
5. What is the expected output of the following code?

Str = ‘abcdef’

Def fun(s):

Del s[2]

Return s

Print(fun(str))

1. abdef
2. The program will cause a runtime exception
3. acdef
4. abcef
5. What is the expected behavior of the following code?

Def f(n):

For i in range(1, n+1):

Yield i

For i in f(2):

Print(i, end=’’)

It will:

1. Print 1 2
2. Print <generator object f at (some hex digits) >
3. Print 2 1
4. Cause a runtime exception
5. Select all the statements ( select two answers)
6. \_\_module\_\_ is a function which returns the name of the module containing the class definition
7. \_\_bases\_\_ is a function filled with the names of the direct subclasses
8. the\_\_name\_\_ attribute exists only inside classes
9. the type() function is able to find the class used to instantiate an object
10. What is the expected output of the following cide?

Def fun(x):

Return 1 if x % 2 !=0 else 2

Print(fun(fun(1))

1. None
2. The code will cause a runtime exception
3. 1
4. 2
5. An interpreter is a program designed to ( select two answers)
6. Check the source code in order to see if it’s correct
7. Execute the source code on the fly
8. Rearrange the source code to make it clearer
9. Translate the source code into machine code
10. The following expression

3 // 2 + 6 / 3

Is equal to:

1. 3.5
2. 0.125
3. 3.0
4. 1.5
5. Assuming that the following snippet has been successfully executed, which of the equations are true? (Select two answers)

a = [1]

b = a

a[0] = 0

1. a[0] + 1 == b[0]
2. a[0] == b[0]
3. b[0] + 1 == a[0]
4. len(a) == len(b)
5. Which of the equations are true? (select two answers)
6. chr(chr(x)) == x
7. ord(chr(x)) == x
8. ord(ord(x)) == x
9. chr(ord(x)) == x
10. What should the initial value of the Val variable be to make the snippet output 16.0?

Val = ???

Val +=1

Val /= 1

Print(Val)

1. 2
2. 3
3. 1
4. 4

1. What is the expected output of the following snippet if the user enters a number equal to 12?

Val = int(input(“Enter value :”))

If val % 2 ==0:

Val +=2

If val %7 ==0:

Val += val // 2

Else:

Val -= val //2

Print(val)

1. 6
2. 21
3. 14
4. 18
5. What is the expected output of the following snippet?

Number =1

Text = “””

Number = number + 1

“””

Number \*=2

Print(number)

1. 1
2. 4
3. 2
4. 0
5. What is the expected behavior of the following snippet?

Y = ‘abc’

Def x():

Print(y, end = ‘’)

Y = ‘cba’

Print(x())

It will:

1. Print cbaabc
2. Cause a runtime exception
3. Print an empty line
4. Print abccba
5. Which of the following print () function invocations is valid?
6. Print(‘line’, line=’’)
7. Print(‘line’,separ=’’)
8. Print(‘line’,end=’’)
9. Print(‘line’,eof=’’)

1. What is the expected output of the following snippet?

I =5

While I > 1:

While I % 2 !=0:

I -=1

Else:

I -=1

Print(i)

1. 2
2. The code is erroneous
3. 0
4. 1
5. What is the expected output of the following snippet?

S = “DON’T JUDGE A BOOK BY ITS COVER”

Print(s.capitalize())

a) DON’T JUDGE A BOOK BY ITS COVER

b) don’t judge a book by its cover

c) Don’t Judge A Book By Its Cover

d) Don’t judge a book by its cover

1. What is the expected behavior of the following snippet?

def fun(a, b=0, c=5, d=1):

return a \*\* b \*\* c

print(fun(b=2, a=2, c=3))

It will:

1. Print 5
2. Print 64
3. Print 256
4. Print 512
5. The code will cause a runtime exception/error
6. What is the expected output of the following snippet?

s = 'python'

for i in range(len(s)):

i = s[i].upper()

print(s, end="")

1. PYTHON
2. Python
3. Python
4. P

Y

T

H

O

N

1. P

y

t

h

o

n

1. The code will cause a runtime exception
2. What is the expected output of the following snippet?

i = 250

while len(str(i)) > 72:

i \*= 2

else:

i //= 2

print(i)

1. 125
2. 250
3. 72
4. 500
5. The following expression

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

is:

a) invalid

b) equal to 16

c) equal to 16.0

d) equal to 512

e) equal to 64

f) equal to 128.0