**Question 1:**

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

**Question 2:**If you want to build a string that reads:

Peter's sister's name's "Anna"

Which of the following literals would you use? (Select all that apply)

A. "Peter's sister's name's \"Anna\""

B. 'Peter\'s sister\'s name\'s \"Anna\"'

C. "Peter's sister's name's "Anna""

D. 'Peter's sister's name's "Anna"'

**Question 3:**

What is the value type returned after executing the following snippet?

x = 0

y = 2

z = len("Python")

x = y > z

print(x)

A. int B. float C. str D. bool  E. NoneType

**Question 4**

Which line can be used instead of the comment to cause the snippet to produce the following expected output? (Select all that apply) Code:

z, y, x = 2, 1, 0

x, z = z, y

y = y - z

**# put line here**

print(x, y, z) Expected output: 0, 1, 2

A. x, y, z = y, z, x B. z, y, x = x, z, y C. y, z, x = x, y, z

D. The code is erroneous

**Question 5:**

What is the expected output of the following snippet?

a = 0 ;b = a \*\* 0

if b < a + 1:c = 1

elif b == 1:   c = 2

else:   c = 3

print(a + b + c)

A. 1 B. 2 C. 3 D. The code is erroneous

**Question 6:**

How many lines does each of the following code examples output when run separately?

**# Example 1**

for i in range(1, 4, 2):

  print("\*")

**# Example 2**

for i in range(1, 4, 2):

 print("\*", end="")

**# Example 3**

for i in range(1, 4, 2):

 print("\*", end="\*\*")

**# Example 4**

for i in range(1, 4, 2):

 print("\*", end="\*\*")

print("\*\*\*")

A. Example 1: two, Example 2: one, Example 3: one, Example 4: one

B. Example 1: two, Example 2: one, Example 3: one, Example 4: two

C. Example 1: two, Example 2: one, Example 3: two, Example 4: three

D. Example 1: one, Example 2: one, Example 3: one, Example 4: two

**Question 7:**

What is the result of the following comparison?

x = "20"

y = "30"

print(x > y)

A. True B. False C. None D. The comparison causes a runtime exception/error

**Question 8:**

What is the expected output of the following snippet?

s = "Hello, Python!"

print(s[-14:15])

A. Hello, Python!  B. !nohtyP ,olleH

C. Hello, Python!Hello, Python!  D. !nohtyP ,olleH!nohtyP ,olleH

E. The program causes a runtime exception/error

F. The result cannot be predicted

**Question 9:**

What is the expected output of the following snippet?

lst = ["A", "B", "C", 2, 4]

del lst[0:-2]

print(lst)

A. [2, 4]

B. ['C', 2, 4]

C. ['B', 'C', 2, 4]  D. ['A', 'B']

**Question 10:**

What is the expected output of the following snippet?

dict = { 'a': 1, 'b': 2, 'c': 3 }

for item in dict:

  print(item)

A. a b c  B. 1 2 3  C. a:1 b:2 c:3  D. 0 1 2  E. The code is erroneous

**Question 11:**

What is the expected output of the following snippet?

s = 'python'

for i in range(len(s)):

  i = s[i].upper()

print(s, end="")

A. PYTHON  B. Python  C. python

D. P

Y

T

H

O

N

E. P

y

t

h

o

n

F. The code will cause a runtime exception

**Question 12:**

What is the expected output of the following snippet?

lst = [i // i for i in range(0,4)]

sum = 0

for n in lst:

  sum += n

print(sum)

A. 0  B. 3  C. 7  D. 11  E. The program will cause a runtime exception

**Question 13:**How many stars (\*) will the following snippet send to the console?

lst = [ [c for c in range(r)]

for r in range(3) ]

for x in lst:

  for y in x:

  if y < 2:

  print('\*', end='')A. One B. Two C. Three D. Four

E. The program will cause a runtime exception/error

**Question 14:**What would you insert instead of ??? so that the program prints 1024 to the  monitor? Code:

lst = [2 \*\* x for x in range(0, 11)]

print(lst**???**)

Expected output: 1024

A. [0]  B. [1]  C. [-1]  D. [1024]  E. [:]

**Question 15:**

What is the expected output of the following snippet?

lst1 = "12,34"

lst2 = lst1.split(',')

print(len(lst1) < len(lst2))

A. True B. False C. None

D. The program will cause a runtime exception/error