PCAP – Certified Associate in Python Programming

Free Sample Test / Practice test

Exam Information

Exam name: **PCAP – Certified Associate in Python Programming**

Exam version: **PCAP-31-02**

Exam duration: **65 minutes** (exam items) + 10 minutes (exam tutorial/Non-Disclosure Agreement)

Number of questions: **40**

Format: **Single-choice and multiple-choice questions**

Passing score: **70%** (28/40 points)

Exam item weight: each question is worth **1 point**

About this Sample Test

This Sample Test covers the following topics:

1. Control and Evaluations (questions 1-10)
2. Data Aggregates (questions 11-20)
3. Functions and Modules (questions 21-30)
4. Classes, Objects, and Exceptions (questions 31-40)

See full Exam Syllabus at https://pythoninstitute.org/pcap-exam-syllabus/

Check out whether you are ready to take your certification exam! The practice questions you find here will give you a good idea of what kind of questions will appear on the real PCAP – Certified Associate in Python Programming exam that you can sit at a Pearson VUE Test Center.

PCAP-31-02 Exam Information and Registrations: https://home.pearsonvue.com/PythonInstitute

Contents

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 expected output of the following snippet?

```
i = 250

while len(str(i)) > 72:

    i *= 2

else:

    i //= 2

print(i)
```

A. 125
B. 250
C. 72
D. 500

Question 4

What snippet would you insert in the line indicated below:

```
n = 0
while n < 4:
    n += 1
    # insert your code here
```

to print the following string to the monitor after the loop finishes its execution:

```
1 2 3 4
```

    A. `print(n)`
    B. `print(n, sep=" ")`
    C. `print(n, end=" ")`
    D. `print(n, " ")`

Question 5

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 6

What will the final value of the `Val` variable be when the following snippet finishes its execution?

```
Val = 1

Val2 = 0

Val = Val ^ Val2

Val2 = Val ^ Val2

Val = Val ^ Val2
```

A. 0
B. 1
C. 2
D. 4
E. The code is erroneous

Question 7

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 8

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 9

How many stars (*) does the following snippet print?

```
i = 10
while i > 0 :
    i -= 3
    print("*")
    if i <= 3:
        break
else:
    print("*")
```

    A. three
    B. two
    C. one
    D. The code is erroneous

Question 10

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 11

Which of the following statements are <u>true</u>? (Select all that apply)

    A. UNICODE is the name of an operating system
    B. UTF-8 is the name of a data transmission device
    C. ASCII is an acronym for *Automatic Systems of Computer Inner Interoperability*
    D. *The Python Language Reference* is the official reference manual that describes the syntax and semantics of the Python language
    E. Python strings are immutable, which means they cannot be sliced
    F. Python strings are mutable, which means they can be sliced
    G. Lists and strings in Python can be sliced

Question 12

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 13

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 14

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 15

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 16

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 17

What is the expected output of the following snippet?

```python
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 18

How many stars (*) will the following snippet send to the console?

```python
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 19

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 20

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

Question 21

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:

A. print `5`
B. print `64`
C. print `256`
D. print `512`
E. The code will cause a runtime exception/error

Question 22

What is the expected behavior of the following snippet?

```
x = 5

f = lambda x: 1 + 2

print(f(x))
```

It will:

A. print 5
B. print 8
C. print 3
D. The code will cause a runtime exception/error

Question 23

What can we deduce from the following snippet? Select the <u>true</u> sentences. (select all that apply)

```
from math import pi as xyz     # line 01

print(pi)                      # line 02
```


    A. The program will print the mathematical constant п = 3.141592…, to available precision

    B. The program will cause a runtime exception/error

    C. The program makes an alias for the name `pi` in the form of `xyz`

    D. The original name `pi` will become inaccessible

    E. Replacing line 02 with `print(xyz)` will cause the program to run without errors

Question 24

What is <u>true</u> about the `__init__.py` file? (Select all that apply)

    A. It cannot be an empty file

    B. It can execute an initialization code for a package

    C. It is required to make Python treat a given directory as a Python package directory

    D. It is required to make Python treat a given directory containing packages as a directory without packages

Question 25

What is the expected behavior of the following code snippet?

```
from random import randint


for i in range(10):

    print(random(1, 5))
```

A. The program will generate a sequence of ten (pseudo)random integers from 1 to 5
B. The program will generate a sequence of ten (pseudo)random integers from 1 to 4
C. The program will generate a sequence of ten (pseudo)random numbers from 1 to 5
D. The program will generate a sequence of ten (pseudo)random numbers from 1 to 4
E. The result cannot be predicted
F. The program will cause a runtime exception/error

Question 26

What is the expected behavior of the following snippet?

```
x = 1               # line 1
def a(x):           # line 2
    return 2 * x    # line 3
x = 2 + a(x)        # line 4
print(a(x))         # line 5
```

It will:

    A. print 8
    B. print 4
    C. print 6
    D. cause a runtime exception on line 4
    E. cause a runtime exception on line 5

Question 27

What is the expected behavior of the following snippet?

```
a = 'hello'      # line 1
def x(a,b):      # line 2
    z = a[0]     # line 3
     return z    # line 4
print(x(a))      # line 5
```

It will:

    A. print `hello`
    B. print `h`
    C. print `ello`
    D. cause a runtime exception on line 2
    E. cause a runtime exception on line 3
    F. cause a runtime exception on line 4
    G. cause a runtime exception on line 5

Question 28

What is the expected behavior of the following snippet?

```
s = 'SPAM'
def f(x):
    return s + 'MAPS'
print(f(s))
```

A. It will print: `SPAM`

B. It will print: `MAPS`

C. It will print: `None`

D. It will print: `SPAMMAPS`

E. It will print: `SPAM MAPS`

F. It will cause a runtime exception/error

G. It will print an empty line

Question 29

Select the <u>true</u> statements:
(select all that apply)

    A. *Positional arguments* are also called *keyword arguments*
    B. The order of arguments matters when they are passed positionally
    C. The order of arguments matters when they are passed by their name
    D. A function can be called with a mix of positional and keyword arguments

Question 30

What is the expected behavior of the following snippet?

```
def gen():
    lst = range(5)
    for i in lst:
        yield i*i


for i in gen():
    print(i, end="")
```

A. It will print: `<generator object gen at` *(some hex digits)*`>`

B. It will print: `014916`

C. It will print:
```
0
1
4
9
16
```

D. It will cause a runtime exception/error

E. It will print an empty line

Question 31

Select the <u>true</u> statements:
(select all that apply)

    A. The `class` keyword marks the beginning of the class definition

    B. An object cannot contain any references to other objects

    C. A class may define an object

    D. A constructor is used to instantiate an object

    E. An object variable is a variable that is stored separately in every object

Question 32

Select the <u>true</u> statements:
(select all that apply)

    A. Inheritance means passing attributes and methods from a superclass to a subclass

    B. `issubclass(class1, class2)` is an example of a function that returns `True` if `class2` is a subclass of `class1`

    C. Multiple inheritance means that a class has more than one superclass

    D. Polymorphism is the situation in which a subclass is able to modify its superclass behavior

    E. A single inheritance is always more difficult to maintain than a multiple inheritance

Question 33

Select the <u>true</u> statements:
(select all that apply)

    A. A class definition may have any number of constructors, but their names must be unique

    B. It is not possible to safely check if an object has a certain attribute

    C. A class constructor cannot return a value

    D. `__bases__` is a tuple filled with the names of all the direct superclasses

    E. `issubclass(c1, c2)` is a function that checks if `c1` is an object derived from class `c2`

Question 34

What will happen when you run each of the following code snippets?

```
# Example 1
x = 1
y = 0

z = x%y
print(z)


# Example 2
x = 1
y = 0

z = x/y
print(z)
```

- A. A `ZeroDivisionError` exception will be raised in Example 1, while Example 2 will print `0` to the screen
- B. A `ZeroDivisionError` exception will be raised in Example 2, while Example 1 will print `0` to the screen
- C. A `ZeroDivisionError` exception will be raised in Example 1 and Example 2
- D. A `ValueError` exception will be raised in Example 1, and a `ZeroDivisionError` exception will be raised in Example 2
- E. A `ValueError` exception will be raised in Example 2, and a `ZeroDivisionError` exception will be raised in Example 1

_____

Question 35

What is the expected output of the following code?

```
x = 0

try:

    print(x)

    print(1 / x)
except ZeroDivisionError:

    print("ERROR MESSAGE")
finally:

    print(x + 1)
print(x + 2)
```

The program will print the following to the screen:

A. 1
   2

B. ERROR MESSAGE
   1
   2

C. 0
   2

D. 0
   ERROR MESSAGE
   1
   2

Question 36

The following class hierarchy is given. What is the expected output of the code?

```
class A:

    def a(self):

        print("A", end='')


class B(A):

    def a(self):

        print("B", end='')


class C(B):

    def b(self):

        print("B", end='')


a = A()

b = B()

c = C()


a.a()

b.a()

c.b()
```

    A. AB
    B. ABB
    C. BA
    D. BBA
    E. AAA
    F. BBB

Question 37

If the following snippet is executed and the exception is raised

```python
try:
    print("Hello")
    raise Exception
    print(1/0)
except Exception as e:
    print(e)
```

we will see:

    A. two identical non-empty lines
    B. two different non-empty lines
    C. two empty lines
    D. one non-empty line and one empty line

Question 38

Is there any difference in the error messages displayed once the two newly defined exceptions `CriticalError` have been raised separately?

```
# Example 1
class CriticalError(Exception):

    def __init__(self, message='ERROR MESSAGE A'):

        Exception.__init__(self, message)


raise CriticalError

raise CriticalError("ERROR MESSAGE B")


# Example 2
class CriticalError(Exception):

    def __init__(self, message='ERROR MESSAGE A'):

        Exception.__init__(self, message)


raise CriticalError("ERROR MESSAGE B")
```

A. No, both errors raised will display the same message: `ERROR MESSAGE A`
B. No, both errors raised will display the same message: `ERROR MESSAGE B`
C. No, both errors raised will display no message
D. Yes, the first error raised will display the message `ERROR MESSAGE A`, while the second `ERROR MESSAGE B`
E. Yes, the first error raised will display no message, while the second `ERROR MESSAGE B`
F. Yes, the first error raised will display no message, while the second `ERROR MESSAGE A`

_____

Question 39

You want to access the `test.txt` file and retrieve each line in it. Which option will you use? (Select all that apply)

```
file = open(test.txt)

# insert code here

file.close()
```

A. `print(file.readlines())`

B. `print(readlines(file))`

C. `print(file.readlines(:)`

D. `for l in file:`
      `print(l)`

E. `print(file.lines())`

F. `print(file.read())`

G. `print(read.file(test.txt))`

Question 40

The following code snippet when run

```
f = open("file.txt", "w")
f.close()
```

will (select all that apply):

    A. open the file `file.txt` in write mode
    B. delete the file contents if the file `file.txt` already exists
    C. leave the file contents unchanged if the file `file.txt` already exists
    D. create the file `file.txt` if it does not exist
    E. raise the `FileNotFoundError` exception if the file does not exist

Answer Key

1. D
2. A, B
3. A
4. C
5. D
6. A
7. A, B
8. C
9. A
10. A
11. D, G
12. B
13. A
14. A
15. A
16. C
17. E
18. C
19. C
20. B
21. C
22. C
23. B, C, D, E
24. B, C
25. F
26. A
27. G
28. D
29. B, D
30. B
31. A, C, D
32. A, C, D
33. C, D
34. C
35. D
36. B
37. D
38. D
39. A, D, F
40. A, B, D

_____

Thank you for downloading and using our Free Sample Test!

We're sure that this test will help you get an idea of what kind of questions you can expect in the real exam, as well as show you whether you're ready to attempt certification, or you need to review some specific knowledge areas.

In fact, the idea behind designing this test was that we wanted to prepare a set of practice questions that would be a little more challenging than the actual exam. If you find this test relatively easy, then we believe that you will have a good chance of passing the real exam.

FAQ

Are there any other study materials I can use to prepare for the *PCAP – Certified Associate in Python* certification exam?

Yes. To prepare for the exam, you can use the learning resources we offer through the Cisco Networking Academy program (https://www.netacad.com/courses/programming/pcap-programming-essentials-python – instructor-led courses) as well as through the OpenEDG Education Platform (https://edu.openedg.org – free, online self-study/self-enroll courses: PCAP Part 1 and PCAP Part 2).

Is there any difference in the format of exam items between this test and the real exam?

No, except these three differences:
- the real exam is carried out in the form of computer-based testing, taken in a Pearson VUE test Center of your choice;
- the real exam items normally have fewer answer options to choose from, usually four or five;
- the real exam multiple-choice items have the number of expected answer options indicated, which means that if you're expected to select two answers, you will be explicitly asked to select two answers.

I found a bug in this test. Can I report it?

Yes, please do. If you'd like to submit feedback for bugs, suggested edits, content typos, etc., please contact us at support@pythoninstitute.org. Thank you!

I'm ready to take the *PCAP – Certified Associate in Python* certification exam. How do I register?

It's very easy to register. Create a Pearson VUE web account (see tutorial), then sign in and book your exam (see tutorial).

Thank you!

_____