

PCAP Certification Exam Preparation



**Get 100% Real Exam Questions, Accurate &
Verified Answers and As Seen in the Real Exam**

Georgio Daccache

Contents

Control and Evaluations (25%)

Questions 1-10

Data Aggregates (25%)

Questions 11-20

Functions and Modules (25%)

Questions 21-30

Classes, Objects, and Exceptions (25%) Questions 31-40

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.

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 true? (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?

```
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?

```
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 true 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 $\pi = 3.141592\dots$, 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 true 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 true 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 true 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 true statements: (select all that apply)

- A. Inheritance means passing attributes and methods from a superclass to a subclass
- B. `is subclass (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 true 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 super classes
- E. `is subclass (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

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

Part 2:

Is it possible to safely check if a class/object has a certain attribute?

- A.** yes, by using the hasattr attribute
- B.** yes, by using the hasattr () method
- C.** yes, by using the hassattr () function
- D.** no, it is not possible

Answer : B

The first parameter of each method:

- A.** holds a reference to the currently processed object
- B.** is always set to None
- C.** is set to a unique random value
- D.** is set by the first argument's value

Answer : D

The simplest possible class definition in Python can be expressed as:

- A.** `class X:`
- B.** `class X: pass`
- C.** `class X: return`
- D.** `class X: { }`

Answer : A

If you want to access an exception object's components and store them in an object called e, you have to use the following form of exception statement:

- A. except Exception (e) :**
- B. except e= Exception :**
- C. except Exception as e:**
- D. such an action is not possible in Python**

Answer : C

A variable stored separately in every object is called:

- A.** there are no such variables, all variables are shared among objects
- B.** a class variable
- C.** an object variable
- D.** an instance variable

Answer : A

There is a stream named s open for writing. What option will you select to write a line to the stream?

- A.** s. write ("Hello\n")
- B.** write (s, "Hello")
- C.** s.writeln ("Hello")
- D.** s. writeline ("Hello")

Answer : A

You are going to read just one character from a stream called s. Which statement would you use?

A. `ch = read (s, 1)`

B. `ch= s.input (1)`

C. `ch= input (s, 1)`

D. `ch= s.read (1)`

Answer : D

**What can you deduce from the following statement?
(Select two answers) `str= open ("file.txt",
"rt")`**

- A.** str is a string read in from the file named file.txt
- B.** a newline character translation will be performed during the reads
- C.** if file. txt does not exist, it will be created
- D.** the opened file cannot be written with the use of the str variable

Answer : A & D

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

```
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 ( )
```

- A. BB
- B. CC
- C. AA
- D. BC

Answer : **D**

Python's built in function named open () tries to open a file and returns:

- A.** an integer value identifying an opened file
- B.** an error code (0 means success)
- C.** a stream object

D. always None

Answer : A

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
```

A. 1

B. 0

C. 2

D. the variable becomes unavailable

Answer: A

An operator able to perform bitwise shifts is coded as (select two answers)

A. - -

B. ++

C. <<

D. >>

Answer: C & D

What will the value of the i variable be when the following loop finishes its execution?

for i in range(10):

pass

A. 10

B. the variable becomes unavailable

C. 11

D. 9

Answer: D

The following expression:

1+-2

is:

- A. equal to 1**
- B. invalid**
- C. equal to 2**
- D. equal to -1**

Answer: D

A compiler is a program designed to (select two answers)

- A. rearrange the source code to make it clearer
- B. check the source code in order to see if it's correct
- C. execute the source code
- D. translate the source code into machine code

Answer: C & D

100 Python Certification Exam Questions Answers to Clear your Test

1. Draw a comparison between Java & Python

Parameters

Java

Python

Ease of use	Good	Very good
Coding speed	Average	Excellent
Data type	Static	Dynamic
Data science & machine learning application		

2. What is Python?

Python is a high-level, general-purpose programming which is simple to interpret, highly interactive and object-oriented scripting language. Python source code is available under General purpose license (GNU). Python is easy to read and uses English keywords frequently. Python has very few syntactical constructions than other languages.

3. Which command is used to exit help window or help command prompt?

You should type quit at the help's command prompt. The help window will automatically get closed and the Python shell prompt will appear.

4. Does the functions help() and dir() list the names of all the built-in functions and variables? If no, how would you list them?

The help() and dir() functions do not list all the names of built-in functions such as max(), min(), filter(), map(), etc. The help() and dir() are available as part of the standard module. To view these functions, you can run the module's builtins as

an argument to “dir()”. It will display the built-in functions, exceptions, and other objects as a list.

5. How do Python check Compile-time and Run-time code?

Python performs most of the checks after the code execution. However, some amount of compile-time check is done before code execution. If a user-defined code is referenced by Python that does not exist, the code gets compiled successfully. In fact, the code will fail with an exception only when the code execution path references the function which does not exist.

6. List the features of python

- Python supports functional programs, and structured programming methods as well as OOP.
 - You can use Python as a scripting language or can be compiled to byte-code for building large applications.
 - Python supports dynamic data types and supports dynamic type check.
 - Python supports automatic garbage collection.
 - You can integrate Python with C, C++, COM, ActiveX, CORBA, and Java very easily.
-

7. Describe the function of PYTHONPATH

PYTHONPATH has a role similar to PATH. This is an environmental variable and informs the Python interpreter about the location of the module files to be imported into a program. Pythonpath includes the Python source library directory and the directories containing Python source code. Python installer sometimes presets the PYTHONPATH.

8. Explain about Python's parameter passing mechanism?

All the parameters pass “by reference” to the functions by default in Python. If the value of the parameter is changed within a function, the change gets reflected in the calling function. When you run the arguments to functions, you can observe the pass “by value” behavior. This is because of the immutable nature of them.

9. Explain the features of Python's Objects.

- The Python's Objects are instances of classes. They are created at the time of instantiation. Eg: object-name = class-name(arguments)
 - More than variable can reference the same object in Python
 - Every object holds exceptional id and it can be obtained by using id() method.
 - Every object can be either mutable or immutable depending on the type of data they hold.
 - When an object is not being used in the code, the following instances can happen automatic destruction, collected as garbage or destruction.
 - You can convert the content of the objects into string representation using a method
-

10. What is the purpose of PYTHONSTARTUP?

PYTHONSTARTUP is an environmental variable. You can specify the location of the file path to a Python file with source code. Every time you run the interpreter, PYTHONSTARTUP gets executed. It is named as .pythonrc.py in Unix.

PYTHONSTARTUP contains commands that load utilities or modify PYTHONPATH.

11. What is the purpose of PYTHONCASEOK?

PYTHONCASEOK is used in Windows to instruct Python to find the first case-insensitive match in an import statement. You can set this variable to any value to activate it.

12. What is the purpose of PYTHONHOME?

The location at runtime is defined by PYTHONHOME. This is an alternative module search path. This module points to the standard library by default. You can switch between module libraries easily.

13. What is Web Scrapping? How do you achieve it in Python?

Web Scrapping is a process of extracting huge data available on websites and storing it in local machines or in the database.

Process:

- Load the web page using requests module.
- Paste the HTML from the web page to get the interesting information.

Python uses few modules for scraping the web. They are urllib2, scrapy, pyquery, BeautifulSoup, etc.

14. What is a Python module?

A Python module is a script that contains import statements, functions, classes, variable definitions, and Python runnable code. This module has a file with an extension `‘.py’`. Modules can also be zip files and DLL files.

The module name can be referred as a string that is stored in the global variable name.

A module can be imported by other modules by `import` or `from module-name import`

15. Name the File-related modules in Python?

Python provides libraries/modules with functions help you to manipulate text files and binary files on file system. With the help of these modules, you can create files, update their content, copy, and delete files. The libraries are `os`, `os.path`, and `shutil`.

- The `os` and `os.path` modules include functions for accessing the filesystem
 - `Shutil` – module helps you to copy and delete the files.
-

16. What kind of data types are supported by Python?

Python supports five data types:

- Numbers
 - String
 - List
 - Tuple
 - Dictionary
-

17. Explain the use of with statement?

A “With” statement is often used to open a file, process the data present in the file, and to close the file without calling a close () method. Python makes exception handling simple with “with” statement and provides cleanup activities.

18. Explain all the file processing modes supported by Python?

You can open Python files in one of the three modes. They are: read-only mode, write-only mode, read-write mode, and append mode by specifying the flags “r”, “w”, “rw”, “a” respectively.

19. Explain how to redirect the output of a python script from stdout (ie., monitor) on to a file?

There are two possible ways of redirecting the output from stdout to a file.

1. You can open an output file in “write” mode. Print the contents into that file, using sys.stdout attribute.

Import sys

Filename = “outputfile” sys.stdout = open () print “testing”

2. You can create a python script say .py file with the contents, say print “testing” and then redirect it to the output file while executing it at the command prompt.

Eg: redirect_output.py has the following code:

Print “Testing”

Execution: python redirect_output.py > outputfile.

20. What is the shortest way to open a text file and display its contents?

The shortest way to open a text file is by using “with” command.

This is done as shown:

With open (“file-name”, “r”) as fp:

```
fileData = fp.read()
```

```
#to print the contents of the file print (fileData)
```

21. How do you create a dictionary which can preserve the order of pairs?

Python dictionaries contain <key, value> pairs and this order should be preserved. They do not preserve the insertion order of <key, value> pairs.

Python 2.7. Introduced a new “OrderDict” class in the “collections” module”. This “OrderDict” provides the same interface like the general dictionaries. It navigates through keys and values in an ordered manner depending on when a key was first inserted.

22. When is a dictionary used instead of a list?

Dictionaries are the best option when the data is labeled, i.e., a record with field names.

Lists are a better option to store collections of un-labeled data like all the files and subdirectories in a folder.

A Search operation on dictionary object is faster than searching a list object.

23. What is the use of enumerate() in Python?

You can iterate through the sequence and retrieve the index position and its corresponding value at the same time using the enumerate() function

24. What are the different kinds of sequences supported by Python? What are they?

Python supports 7 sequence types. They are str, list, tuple, unicode, bytearray, xrange, and buffer.

25. Explain how to perform pattern matching in Python?

You can specify Regular Expressions/REs/ regexes that can match specific “parts” of a given string. The Python’s “re” module provides regular expression patterns and got introduced from later versions of Python 2.5. “re” module provides methods for search patterns for text strings and pattern which are both Unicode strings and 8-bit strings. The re-expression can contain both text characters and regular characters.

26. List the methods for matching and searching the occurrences of a

pattern in a given text String.

The four methods to perform pattern matching in “re”:

Match() – matches the regular expression pattern only to the beginning of the String and not to the beginning of each line.

Search() – scans the string and look for a location where the regular expression pattern matches.

Findall() – finds all the occurrences of the match and return them as a list.

Finditer() – finds all the occurrences of the match and return them as an iterator.

27. Explain split(), sub(), subn() methods to modify strings.

To modify the strings, Python’s “re” module provides 3 methods. They are:

split() – uses a regex pattern to “split” a given string into a list.

sub() – locates all substrings where the regex pattern matches and then different string is replaced.

subn() – This method is similar to a sub() and it returns the new string along with the no. of replacements.

28. How do you display the contents of the text file in reverse order in Python?

1. Convert the given file into a list.
 2. Reverse the list by using reverse ()
-

29. What is JSON? How to convert JSON data into Python data?

JSON – stands for JavaScript Object Notation. It is a common data format for storing data in NoSQL databases. JSON is built on 2 structures:

1. A collection of <name, value> pairs.
2. An ordered list of values.

As Python supports JSON parsers, JSON-based data is actually represented as a dictionary in Python. You can convert JSON data into python using load() of JSON module.

30. Name few Python modules for Statistical, Numerical and scientific calculations.

numPy – provides an array/matrix type computation. It is useful for doing computations on arrays.

scipy – provides methods for doing numeric integrals, solving differential equations

etc pylab – is a module for generating and saving plots

matplotlib – used for managing data and generating plots.

31. What is TkInter?

TkInter is Python library. It is a toolkit to support various GUI tools or widgets for GUI development. The common attributes of them include Dimensions, Colors, Fonts, Cursors, etc.

32. What are the methods are used in the construction and

initialization of custom Objects?

Python uses three methods in the construction and initialization of custom Objects. They are init, new, and del.

new – this method can be considered as a “constructor”.

init — It is an “initializer”/ “constructor” method. It is invoked whenever any arguments are passed at the time of creating an object.

del- this method is a “destructor” of the class. Whenever an object is deleted,

invocation of del__ takes place and it defines behavior during the garbage collection.

33. Is Python object-oriented?

What is object-oriented programming?

Yes. Python is Object Oriented Programming language. OOP is the programming is based on classes and instances of those classes called objects. The features of OOP are:

Encapsulation, Data Abstraction, Inheritance, Polymorphism.

34. What is a Class? How do you create it in Python?

A class is a blueprint/ template of code /collection of objects that has the same set of attributes and behavior. You can create a class using the keyword class followed by class name beginning with an uppercase letter.

35. What is Exception Handling? How do you achieve it in Python?

In Python, an exception is an error that may incur while executing a program. In such a situation, Python generates exception handling, which prevents a program from crashing. Exceptions are special conditions in a program. You can use exception handling when there is a code throwing an error. You can raise an exception by using raise exception statement.

Some keywords useful for exception handling are:

try – A try clause can have many except clauses to handle exceptions to handle differently in different situations. However, it will execute the only one when the exception occurs.

else - The else clause is used to execute a code, when there is no exception raised. It is appropriate to use the else clause than adding code to try clause. The else clause avoids unintentional catching of code.

except – An except clause catches all errors or catches a specific error. It is used after the try block. You can handle multiple exceptions using “except” clause. These exceptions are passed to the clause as “tuple”

print(“incompatible operand types to perform sum”)

raise – The raise clause allows the programmer to force an error to occur

finally – The final clause is executed when an exception occurs no matter what happens. This clause is used to release external sources.

36. Explain Inheritance in Python with an example.

Inheritance is a powerful feature in OOPS and in Python. This feature defines a new class with or without any modification to an existing class. The defined class is called new class or derived class or child class and the one from which it inherits is called base or parent class or superclass. Inheritance gives a code reusability and makes a creation and maintenance of an application easy.

The different types of inheritance supported by the Python are:

Single, multi-level, hierarchical and multiple inheritances.

Single Inheritance – This is a situation where a derived class acquires the members of a single superclass.

Multi-level inheritance – a derived class d1 is inherited from base class base1, and d2 are inherited from base2.

Hierarchical inheritance – You can inherit any number of child classes from one base class.

Multiple inheritances – a derived class is inherited from more than one base class.

37. What is a thread? What is a multithread?

Threading feature is provided by an operating system.

Threading is lighter than processes and shares the same memory space. Threading is a very popular approach to attain concurrency and parallelism. A thread has a beginning, an execution sequence, and a conclusion.

Multi-threading share the same data space within the same process. They can share data, communicate effectively, then when they were a separate process.

38. What are the common exceptions in Python?

The common exceptions in Python are:

- **IOError:** This error happens if the file cannot be opened.
 - **ImportError:** This error happens when python cannot find the module
 - **ValueError:** This error occurs when a built-in operation or function receives an argument that has the correct type but a wrong value
 - **KeyboardInterrupt:** This error occurs if you stop the CPython program by using the interrupt key (normally Control-C or Delete). The Python interpreter immediately throws up KeyboardInterrupt error.
 - **EOFError:** The end of file statement error occurs when one of the built-in functions (`input()` or `raw_input()`) hits an end-of-file condition (EOF) without reading any data. This means that there was open parenthesis on a line which is not matching closing parenthesis. Hence the EOF error.
-

39. Explain about the Overriding method in Python.

The ability of a class to change the implementation of a method provided by one of its ancestors is overriding. This feature is very important in Object-oriented programming like Python as it makes inheritance to get exploited to the fullest potential. Overriding may copy a class without duplicating the code. This feature enhances or customizes a part of the code. Overriding is certainly an integral part of the inheritance. Inheritance is a very powerful method in OOP. It works through implicit delegation.

40. Which methods of Python are used to determine the type of

instance and inheritance?

There are two built-in functions that work with inheritance in Python.

1. `isinstance()` – this method checks the type of instance.
2. `issubclass()` – this method checks class inheritance

`Issubclass(unicode, str)` – returns False because “unicode” is not a subclass of “str”.

41. What are the two built-in functions which come on the keyboard by default to read a line of text from standard input?

The built-in functions to read a line of text from standard input, which comes from the keyboard by default are:

`Raw_input` and `input`

42. How is Python executed?

The files in Python are compiled in bytecode and then executed by the host. Alternatively, type `python.py` at the command line.

43. What is namespace in Python? What are local and global namespaces?

In Python, every name introduced has a place where it resides can be found. This space is known as a namespace. The namespace is an address location, where variable name and

object are mapped. When you search a variable, the address location also gets searched, to get a corresponding object.

Local namespaces are created within a function when that function is called.

When a program starts, then global namespaces are created.

44. How to find undefined g++ symbols `__builtin_new` or `__pure_virtual`?

You can load g++ extension modules dynamically. You have to:

- Recompile and re-link using g++ in Python(change LINKCC in the python Modules Makefile)
 - Link the extension module using g++
-

45. Differentiate between .py and .pyc files?

In Python, both .py and .pyc files hold the bytecode. .py files are Python source files. .pyc are the compiled bytecode files, generated by Python compiler.

46. Explain how to retrieve data from a table in MySQL database through Python code?

1. Import MySQLdb module as `import MySQLdb`
2. Establish a connection to the database as follows:

```
db = MySQLdb.connect("host"="localhost", "database-  
user"="user-name", "password"="password", "database-
```

```
name”=”database”)
```

3. Initialize the cursor variable upon the established connection: `c1 = db.cursor()`
 4. Retrieve the information by defining a required query string.
`s = “Select * from dept”`
 5. Fetch the data using `fetch()` methods and print it. `data = c1.fetch(s)`
 6. Close the database connection. `db.close()`
-

47. What are ODBC and Python?

ODBC (“Open Database Connectivity”) API standard allows the connections with any database that supports the interface, such as PostgreSQL database or Microsoft Access in a transparent manner. There are 3 ODBC modules for Python:

1. PythonWin ODBC module – limited development
 2. mxODBC – a commercial product
 3. pyodbc – it is an open source Python package.
-

48. Define the function to randomize the items of a list in-place?

Python has a built-in module called `random`. This module exports a public method `shuffle()` which can randomize any input sequence.

```
import random
list = [2, 18, 8, 4]
print “Prior Shuffling – 0”, list
random.shuffle(list)
print “After Shuffling – 1”, list
random.shuffle(list)
print “After Shuffling – 2”, list
```

49. Mention how to remove duplicates from a list?

- a. sort the list
 - b. scan the list from the end.
 - c. delete all the duplicate elements from the list when you scan from right-to-left,
-

50. Differentiate between append() and extend() methods. ?

append() and extend() are list methods. Both append() and extend() methods add the elements at the end of the list.

append(element) – This list method adds the given element at the end of the list.

extend(another-list) – adds the elements of another list at the end of the list which is called the extend method.

51. Name few Python Web Frameworks for developing web applications?

There are various web frameworks provided by Python. They are:

web2py – This is the simplest of all the web frameworks used for developing web applications.

cherryPy – This is an Object-oriented Web framework in Python.

Flask – This is a micro-framework for designing and developing web applications in Python.

52. How do you check the file existence and their types in Python?

`os.path.exists()` – This method is used to check the existence of a file. Python returns True if the file exists, otherwise false

`os.path.isfile()` – This method is used to check if the given path references a file or not. Python returns True if the path references to a file, else it returns false.

`os.path.isdir()` – This method is used to check whether the given path references a directory or not. It returns True if the path references to a directory, else it returns false.

`os.path.getsize()` – returns the size of the given file

`os.path.getmtime()` – returns the time of the given path when last accessed. The value returned is a number.

53. Mention a few methods that are helpful in the implementation of Functionally Oriented Programming in Python?

Some methods supported by Python (called iterators in Python3), such as `filter()`, `map()`, and `reduce()`, are very useful when you need to repeat over the items in a list, create a dictionary, or extract a subset of a list.

`Filter()` – helps you to extract a subset of values based on conditional logic.

`map()` – it is a built-in function that applies the function to each item in an iterable.

`reduce()` – iterates a pair-wise reduction on a sequence for single value computation.

54. What is Python dictionary?

Python's dictionaries map unique keys to values. They are associative arrays or hashes. Dictionaries are mutable. This means they are changeable. A dictionary key can be any Python type but are usually numbers or strings, Values, integers, tuples, any arbitrary Python object.

55. How do you retrieve values from the dictionary?

You can get all the values from dictionary object using a `dictionary.values()` function.

56. How to get keys from the dictionary?

You can get all the keys from dictionary object using a `dictionary.keys()` function.

57. What are the various numeric types in Python?

Python has four different numeric types.

1. `int` (plain integers): Integers are standard positive or negative whole numbers.
2. `long` (long integers): long integers are of infinite size. They are just like plain integers followed by the letter "L" (ex: 150L).
3. `float` (floating point real values): floats represent real numbers. They are written with decimal points to divide the whole number into fractional parts.
- `complex` (complex numbers): Complex is represented by the formula $a + bJ$, where a and b are floats, and J is the

square root of -1 (the result of which is an imaginary number). Complex numbers are very rarely used in Python.

58. What are tuples?

A sequence of immutable Python objects is known as tuples. Tuples are just like lists except that tuples cannot be changed and they use parentheses. A tuple can be created by using different comma separated values.

59. Difference between tuple and list.

The list is mutable and the tuple is immutable. In a tuple, a fixed memory is assigned to the variable. In a list, more memory is used than actually used. The List is enclosed in brackets, a tuple is enclosed in parentheses.

60. How to get the length of a list?

The length of the list is denoted by `len (list)`.

61. How to get the max value of the list?

`Max(list)` function returns the item from the list with max value.

62. How to get min value of the list?

Min(list) function returns the item from the list with a min value.

63. How to get the index of an object in a list?

list.index(obj) function returns the lowest index in the list where the object appears.

64. When do you use a continue statement in a for loop?

A continue statement in a for loop is used, when the processing for a particular item is completed to move on to the next. The continue statement states, that the current item's processing is done and ready to move on to the next item.

65. When is a break statement used in a for loop?

The break statement gives you an indication that the function of the loop is completed and you should move to the next block of code. For example, when the item being searched is found, there is no need to keep looping. The break statement plays its role here and the loop terminates. The execution moves on to the next section of the code.

66. What is PEP8?

PEP 8 is a coding convention which helps to make Python code more readable.

67. How is Python language interpreted?

Python is an interpreted language. This language runs directly from source code. The programmer writes the source code which gets converted into intermediate language, which is translated into machine language which is executed.

68. Name the tools that identify the bugs and perform static analysis.

Pychecker is a static analysis tool. This tool detects bugs from Python source code. It also cautions about style and complexity of the bug.

Pylint tool verifies whether the module meets the coding standard.

69. Explain about Python decorators?

A Python decorator is a specific change made in the Python syntax to alter functions easily.

70. What is lambda in Python?

A lambda is a tool for creating a single expression, one-time, small anonymous function objects in Python. A lambda makes new function object and returns at runtime. It can have any number of arguments. Lambda functions are used in combination with filter(), map(), and reduce() functions.

71. What is slicing?

Slicing is a selection of a range of items from sequence types like list, tuple, strings etc.

72. What are iterators?

Iterators are used to iterate a group of elements, containers etc.

73. What is unit test?

In Python, a unit testing framework is known as unit test.

74. What are generators?

The method of implementing iterators is known as generators. It is a normal function yielding expression in the function.

75. What is docstring?

A python documentation string is called docstring. Docstring documents python functions, modules, and classes.

76. How to copy an object?

You can use `copy.copy()` or `copy.deepcopy()` for the general case. You can copy most of the objects but not all of them.

77. Mention the difference between X-Range and Range.

Xrange returns the xrange object. A range returns the list, and uses the same memory and no matter what the range size is.

78. What is a module and package in Python?

A module is a way to structure program. Each Python program file is a module, which imports other modules like objects and attributes. A module is a Python file with .py extension. A package is a namespace which can have modules or subfolders. A package is a simple directory.

79. How can a python script be executed on Unix?

The steps to execute a python script on Unix are:

- Script file's mode must be executable
 - The first line must begin with # (#!/usr/local/bin/python)
-

80. How can you access a module written in Python from C

Follow this step:

```
Module = PyImport_ImportModule("");
```

81. How to delete a file in Python?

Use the command `os.remove(filename)` or `os.unlink(filename)` to delete a file.

82. How to generate random numbers in python?

To generate random numbers, you have to import command:

```
import random
```

`random.random()`

This command will return random floating point number in the range (0,1)

83. What is the use of a //operator in Python?

A // operator is a floor division operator, used for dividing two operands. The quotient shows only digits before the decimal points.

For example: $10//5=2$ and $10.0//5.0=2.0$

84. What is the use of a split function in Python?

A split function breaks a string into shorter strings using a defined separator. It gives a list of all words present in the string.

85. What is a flask and what are its benefits?

A flask is a web microframework and will have no dependencies on external libraries. It makes the framework light with less dependency to update and lowered security bugs.

86. How to minimize the Memcached server outages in your Python Development?

In case of an instance failure, several other instances also go down. Whenever the client makes a request, there will be a load on the database server. To avoid this, you should write a code to minimize cache stampedes. Then, there will be a minimal impact

You can bring up an instance of Memcached on a new machine, by using the lost machines IP address.

To minimize the server outages, the code is another wise option. It gives you the flexibility to change the Memcached server list with less effort.

87. How should Memcached not be used in Python project?

- Memcached should be used as a cache, not as a data store.
 - Use multiple sources for data availability. Never use Memcached as the only source to run your application.
 - Memcached does not provide any security for encryption or authentication.
 - Memcached is a key or value store. It cannot perform any query over data or over the content to extract information.
-

88. What is a dogpile effect? How to prevent?

Whenever the cache expires, the websites are hit by multiple client requests at the same time. You can prevent this effect using semaphore lock. When a value expires, the first process acquires the lock and generates new value.

89. Identify the common way for the flask script to work?

The common ways identified are:

- It should be the import path for your application
 - The path to a python file
-

90. What is pass in python?

An operation in Python statement is a “pass”. In other words, it’s a placeholder in a compound statement where nothing is written. Just a blank is left.

91. How to manage memory in Python?

Python memory is managed by Python private heap space. All the python objects and data structures are located in heap space. The interpreter handles private heap space and the programmer does not have access.

The Python memory manager allocated python heap space for objects. The programmer gets access to some tools for coding with the help of the core API.

All the unused memory gets recycled in in-built garbage collector of Python. This unused memory is available for heap space.

92. What is pickling and unpickling?

The python pickle module is a process where any object is converted into a byte stream. Later dumps into a file using dump function. This process is termed as pickling.

Unpickling is when the byte stream is converted back into objects, The process of retrieving the original python objects from stored string representation is called unpickling.

The other names for pickling and unpickling are serialization, marshalling, or flattening. The pickling process is done to save

the objects in a disc.

93. How many arguments pass by value or reference?

Python has everything in object form. All the variables hold references to the objects. The reference values are according to functions, hence the value of the reference cannot be changed. You can change the objects if it is mutable.

94. What are dict or list comprehensions?

Dict and list comprehensions are syntax constructions to simplify the creation of dictionary.

95. What built-in-type python provides?

Python builds mutable and immutable built-in-types

Mutable:

- List
- Sets
- Dictionary

Immutable built-in types

- Strings
 - Tuples
 - Numbers
-

96. What is a negative index in Python?

The python arrays and list items are accessed with positive or negative numbers. These numbers are called index.

For a negative index, -n is the first index. -(n-1) is the second one. The last negative index is -1. All the elements are assessed by a negative index from the end of the list by backward counting.

97. How do you track different versions of your code?

Version control is the best way to keep track of your code.

98. What is monkey patching?

Changing the behavior pattern of function or object after being defined is known as monkey patching.

99. What does *args and **kwargs mean?

If we are not sure how many arguments will be passed to a function or pass stored list or tuple of arguments, then we use the function*args.

**kwarg is used when we are not sure how many keyword arguments will be passed to a function.

100. What are @classmethod, @staticmethod, @property?

All the three are decorators. A decorator is a function that takes a function and returns a function or takes a class and

returns a class.

101. Explain garbage collection process

A count of the number of references to a given object is maintained. If a reference count goes down to zero, then the object does not exist anymore and the memory gets freed up. The garbage collector looks for “reference cycles” occasionally. It cleans up objects if they are not referenced, as they should not be live. As and when the objects are created, they are assigned to generations. Each object gets one generation. The younger object is dealt first.