

Python test

### 1. Which of the following is a way to find a local computer's IP address with Python?

**1. import socket  
socket.gethostbyname(socket.gethostname())**

2. socket.gethostbyname(socket.gethostname())

3. import socket  
gethostbyname(socket.gethostname())

4. import ifconfig  
print (ifconfig -a)

### 2. Which of the following statements copy the contents of a list and not just a reference to the list?

1. newlist = oldlist

**2. newlist = oldlist[:]**

3. newlist = oldlist(dummy)

4. None of these

### 3. What would the 'sorted\_tel' be in the following code: tel = {'jack': 4098, 'sape': 5139, 'bill': 3678, 'mike': 2122} sorted\_tel = sorted(tel.items(), key=lambda x: x[1])

1. A dictionary sorted by the second element.

**2. A list of Tuples sorted by the second element.**

3. A list of dictionaries sorted by the first element.

4. A dictionary sorted by the first element.

5. A list of Tuples sorted by the first element.

### 4. Which of the following statements can be used to remove an item from a list by giving the index?

1. remove listname[index]

**2. del listname[index]**

3. kill listname[index]

4. None of these

### 5. Read the following statements: >>> import array >>> a = array.array('c','spam and eggs') >>> a[0] = 'S' >>> a[-4:] = array.array('c','toast') >>> print ''.join(a) Which of the following will be the output of the above code snippet?

**1. Spam and toast**

2. spam and toast

3. Spam and eggs

4. spam and eggs

5. spamandtoast

6. spamandeggs

### 6. Which is the best way to convert a Python time.struct\_time object into a datetime object?

1. structTime = time.localtime()  
datetime.datetime(\*structTime[:6])  
datetime.datetime(2009, 11, 8, 20, 32, 35)

**2. datetime.fromtimestamp(mktime(struct))**

3. A time.struct\_time object cannot be converted into a datetime object.

4. datetime.fromtimestamp(struct)

### 7. For the list "thelist = [1, 2, 3, 4, 5]", how could the last two elements of the list (4, 5) be selected using slices?

**1. thelist[3:]**

2. thelist[3,4]

3. thelist[3:4]

4. thelist[-2]

### 8. Which of the following is the correct way to call the private method, myPrivateMethod(), in class MyClass, using dir(obj)? class MyClass: def \_\_myPrivateMethod(self): print "Private Method" obj = MyClass() print dir(obj)

1. \_\_myPrivateMethod

**2. \_MyClass\_\_myPrivateMethod**

3. myPrivateMethod

4. A private method will not be shown in the output.

### 9. Which function could be used to list every file and folder in the current directory?

1. os.dirnames('.')

**2. os.listdir('.')**

3. os.listdir('/')

4. os.ls()

### 10. Writing to STDOUT and STDERR is fairly inflexible, and most of the time the print statement accomplishes the same purpose more flexibly. How many arguments can a print statement handle?

1. 1

2. 2

3. 7

**4. Any number**

### 11. While running an application, a user pressed the interrupt key (Ctrl + C). Which of the following exceptions will occur?

1. InterruptError

**2. KeyboardInterrupt**

3. KeyboardInterruptError

4. None of these

### 12. Divide by zero is a very common error. Which of the following handles this exception?

1. DivideByZeroError

2. DivisionZeroError

**3. ZeroDivisionError**

4. DivisionError

### 13. Which Python module can be used for copying files?

1. util

**2. shutil**

3. copy

4. filecopy

### 14. What is the correct way to delete a directory that is not empty using Python?

1. os.remove('/directory')

2. os.rmtree('/directory')

3. shutil.rmtree('/directory')

**4. import shutil  
shutil.rmtree('/directory')**

### 15. Which of the following is the correct prototype of the string.find() function?

1. string.find(s, sub ,start ,end)

2. string.find(s, sub ,start [,end])

**3. string.find(s, sub [,start [,end]])**

4. string.find(s, sub [,start] ,end)

### 16. In Python 2.x, which of the following is the way to check to make sure that the variable 'x' is not a string?

1. assert isinstance(x, basestring)

**2. assert not isinstance(x, basestring)**

3. assert not instance(x, basestring)

4. assert not x.isinstance(basestring)

### 17. Read the following code snippet: import types, exceptions def overloaded\_get\_text(o): if type(o) is type.FileType: text = o.read() else: raise exceptions.TypeError return text Which of the following lines has incorrect syntax?

1. if type(o) is type.FileType:

2. text = o.read()

3. raise exceptions.TypeError

**4. None of these**

### 18. Which of the following is the correct way to recursively find files with the extension ".html" in the directory "site" with Python?

1. import fnmatch  
import os  
  
matches = os.walk('site', '\*.html')

2. import fnmatch  
import os  
  
matches = []  
for root, dirnames, filenames in os.walk('site'):  
 for filename in fnmatch(filenames, '\*.html'):  
 matches(os.path.join(root, filename))

**3. import fnmatch  
import os  
  
matches = []  
for root, dirnames, filenames in os.walk('site'):  
 for filename in fnmatch.filter(filenames, '\*.html'):  
 matches.append(os.path.join(root, filename))**

4. matches = []  
for root, dirnames, filenames in os.walk('site'):  
 for filename in fnmatch.filter(filenames, '\*.html'):  
 matches.append(os.path.join(root, filename))

### 19. Which of the following is the best way to reverse the string 'Test String' in Python?

1. string.reverse('Test String')

**2. 'Test String'[::-1]**

3. reversed('Test String')

4. 'Test String'[-1:0]

### 20. Which of the following is the correct method for changing a global variable inside a function?

1. def change\_globvar():  
 globvar = 1

**2. def change\_globvar():  
 global globvar   
 globvar = 1**

3. def change\_globvar():  
 import globvar   
 globvar = 1

4. def change\_globvar():  
 global globvar = 1

### 21. Which of the following modules lets you check whether two files are identical, and whether two directories contain some identical files?

1. dircomp

2. filecompare

**3. filecmp**

4. structcomp

### 22. How can a null object be declared in Python?

1. none

**2. None**

3. null

4. Null

### 23. Select all incorrect statements regarding functions decorated with staticmethod and functions decorated with classmethod:

There may be more than one answer.

1. A classmethod receives the class as its implicit first argument whereas a staticmethod does not receive an implicit first argument.

**2. Syntax to declare a classmethod:  
class C:  
 @classmethod  
 def f(cls, arg1, arg2, ...): ...**

**3. Syntax to declare a staticmethod:  
class C:  
 @staticmethod  
 def f(arg1, arg2, ...): ...**

4. The first argument for a classmethod function must always be class.

5. Staticmethod is callable without instantiating the class first, whereas classmethod is not.

### 24. Read the following statements: Statement 1: Many string module functions are now also available as string object methods. Statement 2: To use string object methods, there is no need to import the string module. Which of the following is correct?

1. Statement 1 is true, but statement 2 is false.

2. Statement 2 is true, but statement 1 is false.

**3. Both statements are true.**

4. Both statements are false.

### 25. Which of the following members of the object class compare two parameters?

There may be more than one answer.

**1. object.\_\_eq\_\_(self, other)**

**2. object.\_\_ne\_\_(self, other)**

3. object.\_\_compare\_\_(self, other)

4. object.\_\_equals\_\_(self, other)

5. object.\_\_co\_\_(self, other)

6. None of these

### 26. What will be the output of the following statements: >>> import string >>> string.ljust(width=30,s="Mary had a little lamb")

**1. 'Mary had a little lamb '**

2. 'Mary had a little lamb'

3. ' Mary had a little lamb'

4. None of these

### 27. Which of the following will disable output buffering in Python?

There may be more than one answer.

**1. Using the -u command line switch**

**2. class Unbuffered:  
 def \_\_init\_\_(self, stream):  
 self.stream = stream  
 def write(self, data):  
 self.stream.write(data)  
 self.stream.flush()  
 def \_\_getattr\_\_(self, attr):  
 return getattr(self.stream, attr)  
  
import sys  
sys.stdout=Unbuffered(sys.stdout)**

**3. Setting the PYTHONUNBUFFERED environment variable**

4. sys.stdout = os.fdopen(sys.stdout.fileno(), 'w', 0)

### 28. Sometimes, an object needs to be converted into its string representation. Fortunately, the object class provides functionality for this. Which of the following returns a string representation of the object 'self'?

**1. object.\_\_repr\_\_(self)**

2. object.\_\_string\_\_(self)

3. object.\_\_conv\_\_(self)

4. object.\_\_slz\_\_(self)

### 29. Object is the base class of new-style datatypes. Which of the following functions is not a member of the object class?

1. object.\_\_eq\_\_(self, other)

2. object.\_\_ne\_\_(self, other)

**3. object.\_\_nz\_\_(self)**

4. object.\_\_repr\_\_(self)

5. None of these

### 30. What is the output of "Dummy Text is my text".strip('Te') -

**1. 'Dummy Text is my text'**

2. 'Dummy xt is my xt'

3. 'Dummy xt is my text'

4. None of these

### 31. Read the following statements: Statement 1: The conditions used in 'while' and 'if' statements can contain only comparison operators. Statement 2: The operators 'is' and 'is not' compare whether two objects are really the same object. Which of the following is correct?

1. Statement 1 is true, while statement 2 is false.

2. Statement 1 is false, while statement 2 is true.

**3. Both statements are true.**

4. Both statements are false.

### 32. If a name has not been bound inside some accessible scope, how can it be dereferenced?

1. Through a global namespace

2. Through package.module

3. Through the normal dereference process

**4. It cannot be dereferenced**

### 33. Which of the following will determine the number of CPUs available in the operating environment?

1. import os  
  
os..cpu\_count()

2. import sys  
  
sys..cpu\_count()

3. import psutil  
  
sutil.NUM\_CPUS

**4. import multiprocessing  
  
multiprocessing.cpu\_count()**

### 34. What is a metaclass in Python?

1. A class that is inherited from

2. A class that inherits from another class

**3. Something that can be attached to any class, that gives it a constant set of attributes**

4. Something that creates "class" objects

### 35. Which of the following is the correct prototype for the 'open' function of the file class in python 2.2+?

**1. open(fname [,mode [,buffering]])**

2. open(fname [,buffering [,mode]])

3. open(fname [,mode])

4. open(fname,mode,buffering)

5. open(fname,buffering,mode)

6. None of these

### 36. Which of the following exceptions is raised while using sys.exit()?

1. SysExit

**2. SystemExit**

3. Exit

4. ExceptExit

### 37. Which is the correct syntax to call the parent class of the \_\_init\_\_ method?

1. super.\_\_init\_\_()

2. super(ParentClass).\_\_init\_\_()

**3. super(ParentClass, self).\_\_init\_\_()**

4. super.ParentClass()

### 38. In Python, built-in exceptions can be inherited from. Which of the following is the base exception class?

1. Exceptions

2. BaseException

**3. Exception**

4. Except

### 39. Which of the following methods returns the ASCII value of a character in Python?

1. ascii

**2. ord**

3. asciicode

4. None of these

### 40. Function sys.getrefcount(obj) returns the number of references to the object obj. Which of the following is the correct value returned by this function?

1. The total number of references

**2. One higher than the total number of references**

3. One lesser than the total number of references

4. Double of the total number of references

### 41. Which of the following is an incorrect way of determining whether the type of an object (obj) is list?

1. type(obj) == list

2. type(obj) is list

**3. isinstance(obj, list)**

4. obj is list

### 42. Which of the following statements regarding tuples and lists is wrong?

1. Lists are mutable while tuples are immutable, and cannot be modified once created.

2. Tuples are hashable and can be used as dictionary keys, lists are not.

**3. Tuples usually contain a heterogeneous sequence of elements that are accessed via unpacking, but not indexing.**

4. Elements of lists are usually homogeneous and are accessed by iterating over the list.

### 43. Read the following statements: >>> word = 'Help' + 'A' >>> word[2:] Which of the following will be the output of the above code?

**1. lpA**

2. He

3. pA

4. lp

### 44. Examine the following prototype for the 'open' function of the file class in Python 2.2+: open(fname [,mode [,buffering]]) Which of the following is correct for the 'buffering' argument?

1. 0 for none

2. 1 for line oriented

3. An integer larger than 1 for number of bytes

**4. All of these**

### 45. Read the following statements: >>> word = 'Help' + 'A' >>> '<' + word\*5 + '>' Which of the following will be the output of the above code snippet?

**1. '<HelpAHelpAHelpAHelpAHelpA>'**

2. '<HelpA5>'

3. '<HelpA\*5>'

4. An error

5. None of these

### 46. Inheriting from a base class enables a custom class to use a few new capabilities, such as slots and properties. Which of the following is the base class of new-style datatypes?

1. base

**2. object**

3. dict

4. custom

5. None of these

### 47. Read the following statements: >>> word = 'Help' + 'A' >>> word[:2] Which of the following will be the output of the above code?

1. lpA

**2. He**

3. pA

4. lp

### 48. What will be the output of the following statement: >>> "ab123".islower(), '123'.islower(), 'Ab123'.islower()

1. (True,True,False)

2. (True,False,True)

**3. (True,False,False)**

4. (False,False,False)

### 49. Which option will import a module which is not in PYTHONPATH or the current directory?

1. import <modulename>

**2. Add the path in program by sys.path.insert(<path>)**

3. <modulename> import \*

4. Put \_\_init\_\_.py file in the module path and import it using the import statement

### 50. Which of the following statements will redirect the output of the print statement into a file-like object named 'test'?

1. print >> open('test','w'), "Pi: %.3f" % 3.1415, 27+11

2. print 'test', "Pi: %.3f" % 3.1415, 27+11

3. print open('test','w'), "Pi: %.3f" % 3.1415, 27+11

4. print $$ open('test','w'), "Pi: %.3f" % 3.1415, 27+11

### 51. Which of the following is correct about Python?

1. Python is a byte-code compiled programming language.

2. Python uses a virtual machine to run machine-abstracted instructions.

3. Python is an object-oriented language.

4. Python is a declarative programming language.

**5. All of these.**

### 52. Which of the following statements are true? A. .\_variable is semi-private and meant just for convention. B. .\_\_variable is considered superprivate and gets name mangled to prevent accidental access. C. .\_\_variable\_\_ is typically reserved for built-in methods or variables.

1. A

2. B

3. C

**4. A, B, and C**

### 53. To make numbers be of a specific length, zeroes are padded to the left. Which of the following is the best way to pad a string with zeroes to the left, to make its length 5? The string is: num = '123'

1. print num.zfill(3)

2. print '%03d' % num

**3. print num.zfill(5)**

4. print '%05d' % num

### 54. Which function prints all of the current properties and values of any kind of object?

There may be more than one answer.

**1. dir(someObject)**

**2. someObject.\_\_dict\_\_**

**3. from pprint import pprint  
pprint (vars(someObject))**

4. import inspect  
inspect.getobjectinfo(someObject)

### 55. Which of the following is the correct way to print a list called theList in reverse order?

**1. for i in reversed(theList):  
 print i**

2. for i in theList[:-1]:  
 print i

3. for i in theList.reversed:  
 print i

4. for i in reversed(theList)[-1:]:  
 print i

### 56. Read the following code snippet: >>> o1 = C() >>> o1.lst = [1,2,3] >>> o3 = copy.deepcopy(o1) >>> o1.lst.append(17) What will be the output of '>>> o3.lst' and '>>> o1.lst'?

1. [1, 2, 3] and [1, 2, 3, 17]

2. [1, 2, 3, 17] and [1, 2, 3, 17]

3. [1, 2, 3] and [1, 2, 3]

**4. None of these**

### 57. Which of the following exceptions occurs while importing a module?

1. ModuleError

**2. ImportError**

3. ImportModuleError

4. ReferenceError

### 58. Which of the following protocol libraries can be used for an email implementation in a Python application?

1. telnetlib

**2. smtplib**

3. ftplib

4. None of these

### 59. Which of the following code samples will read arguments if the script is run from the command line?

1. import argparse  
parser = argparse.ArgumentParser()  
print parser.arguments()

2. from optparse import OptionParser  
parser = OptionParser()  
print parser.arguments()

**3. import sys  
print sys.argv**

4. import os  
print os.argv

### 60. What is the best way to check if the object 'myobject' is iterable in Python?

1. myobject.isIter()

**2. try:  
 [ e for e in myobject]  
except TypeError:  
 print myobject, 'is not iterable'**

3. try:  
 iter.myobject  
except:  
 print myobject, 'is not iterable'

4. if [ e for e in myobject]:  
 print myobject, 'is iterable'  
else:  
 print myobject, 'is not iterable'

### 61. What is the correct syntax that will set the variable "var1" to the value "correct" if variable "a" is greater than 1?

1. var1 = "correct" if a > 1

**2. var1 = "correct" if a > 1 else None**

3. var1 = "correct" if a > 1 else if None

4. var1 = "correct" if a > 1 else

### 62. Which of the following is the correct way to print to stderr in Python?

1. sys.stderr.write('my\_error')

2. print >> sys.stderr, 'my\_error'

3. from \_\_future\_\_ import print\_function  
 print('my\_error', file=sys.stderr)

**4. All of these**

### 63. Various email and news clients store messages in a variety of formats, many providing hierarchical and structured folders. Which of the following provides a uniform API for reading the messages stored in all the most popular folder formats?

1. mailstruct

2. emailarch

3. emailfolders

4. mailbox

### 64. What would be the output of the following code? def change\_list(list1, list2): list1.append('four') list2 = ['and', 'we', 'can', 'not', 'lie'] outer\_list1 = ['one', 'two', 'three'] outer\_list2 = ['we', 'like', 'proper', 'English'] change\_list(outer\_list1, outer\_list2) print outer\_list1 print outer\_list2

**1. ['one', 'two', 'three', 'four']  
['we', 'like', 'proper', 'English']**

2. ['one', 'two', 'three', 'four']  
['and', 'we', 'can', 'not', 'lie']

3. ['one', 'two', 'three']  
['we', 'like', 'proper', 'English']

4. ['one', 'two', 'three']  
['and', 'we', 'can', 'not', 'lie']

### 65. Which of the following code snippets concatenates the list a\_list = [1, 2, 3] with the tuple a\_tuple = (4, 5), so the result would be [1, 2, 3, 4, 5]?

1. a\_list + a\_tuple

**2. a\_list.extend(a\_tuple)**

3. a\_list.append(a\_tuple)

4. a\_list.append(\*a\_tuple)

### 66. Which is the correct way to remove an installed Python package?

1. easy\_install -v <package>

**2. pip uninstall <package>**

3. easy\_install -u <package>

4. pip -m <package>

### 67. What is the output of the following code? def foo(param1, \*param2): print param1 print param2 def bar(param1, \*\*param2): print param1 print param2 foo(1,2,3,4,5) bar(1,a=2,b=3)

1. TypeError: foo() got multiple values for keyword argument 'param1'

2. 1  
(2, 3, 4, 5)  
1  
(2, 3)

**3. 1  
(2, 3, 4, 5)  
1  
{'a': 2, 'b': 3}**

4. 1  
{2, 3, 4, 5}  
1  
('a': 2, 'b': 3)

### 68. Which of the following code snippets converts the hexademical number 0x1000 to a decimal correctly?

**1. int('1000', 16)**

2. int(1000)

3. 1 \* (16 \*\* 4) + 0 + 0 + 0

4. int.fromhex('0x1000')

### 69. In Python, what is the default maximum level of recursion?

1. 500

**2. 1000**

3. 10000

4. There is no default maximum level

### 70. What is the most flexible way to call the external command "ls -l" in Python?

**1. from subprocess import call  
call(["ls", "-l"])**

2. from subprocess import call  
call("ls -l")

3. os.system("ls -l")

4. os.system(["ls"], ["-l"])

### 71. Which of the following modules keep prior directory listings in the memory to avoid the need for a new call to the file system?

1. sys

2. FileSys

3. dirsys

**4. dircache**

### 72. What is the output of the following code: name = 'Jon' name.rjust(4, 'A')

1. 'Jon A'

2. 'A Jon'

**3. 'AJon'**

4. 'JonA'

### 73. Consider the function: def hello(): return "Hello, world!" Which of the following is the correct way to make a decorator that could be used to make hello() return "#Hello, world!#"?

1. def hello():  
 return "#Hello, world!#"

**2. def hashes(fn):  
 def wrapped():  
 return "#" + fn() + "#"  
 return wrapped  
  
@hashes  
def hello():  
 return "Hello, world!"**

3. def hashes(fn):  
 def wrapped():  
 return "#" + "#"  
 return wrapped  
  
@hashes  
def hello():  
 return "Hello, world!"

4. def hashes(fn):  
 def wrapped():  
 print "#" + fn() + "#"  
 return  
  
@hashes  
def hello():  
 return "Hello, world!"

### 74. Python 2.3+ includes a standard module that implements a set datatype. Which of the following statements is true about the set datatype?

1. A set is an unordered collection of hashable objects.

2. No object can occur in a set more than once.

3. A set resembles a dict that has only keys but no values.

4. Sets utilize bitwise and Boolean syntax to perform basic set-theoretic operations.

**5. All of these.**

### 75. What command is used to get the hostname of the computer the Python script is running on?

1. print hostname()

2. import socket  
print gethostname()

3. print socket.gethostname()

**4. import socket  
print socket.gethostname()**

### 76. Which of the following modules can be used for profiling Python code?

1. profile

2. cProfile

3. pyProfile

**4. profile and cProfile**

### 77. Which of the following variables store parameters passed from outside?

1. sys.param

2. sys.arg

**3. sys.argv**

4. sys.prm

### 78. Which of the following functions can change the maximum level of recursion?

1. setmaxrecursion function in the sys module

**2. setrecursionlimit function in the sys module**

3. setmaximumrecursion function in the sys module

4. None of these

### 79. What is the best way to randomly select an item from list?

**1. foo = ['a', 'b', 'c', 'd', 'e']  
from random import choice  
print choice(foo)**

2. foo = ['a', 'b', 'c', 'd', 'e']  
from random import randrange  
random\_index = randrange(0,len(foo))  
print foo[random\_index]

3. foo = ['a', 'b', 'c', 'd', 'e']  
import random  
print int(random.random() \* len(foo))

4. foo = ['a', 'b', 'c', 'd', 'e']  
import random  
print foo[int(random.random() \* len(foo)+1)]

### 80. Which of the following functions modifies the list in place to indicate which items are directories, and which are plain files?

1. dircache.listdir(path, lst)

**2. dircache.annotate(path, lst)**

3. dircache.filter(path, lst)

4. dircache.scan(path, lst)

### 81. How can a method be declared static in Python?

**1. Using the @staticmethod decorator**

2. By declaring static by adding the static keyword before a method

3. A method inside a class without its name as an argument is static method.

4. There are no static methods in Python.

### 82. Which of the following is the correct way to output the name of any function (including builtin functions) using Python?

1. theFunction.func\_name

**2. theFunction.\_\_name\_\_**

3. theFunction.name()

4. theFunction.\_\_func\_name\_\_

### 83. One common way to test a capability in Python is to try to do something, and catch any exceptions that occur. Which of the following is the correct mechanism of trapping an error?

1. try:  
 code to be verified will come here  
exception <except>:

**2. try:  
 code to be verified will come here  
except <exception>:**

3. try:  
 code to be verified will come here  
exception:

4. try:  
 code to be verified will come here  
exception <exception>:

### 84. Which of the following is the correct way to execute a program from inside Python without having to consider how the arguments/quotes are formatted?

1. import subprocess  
subprocess.call('C:\\Temp\\a b c\\Notepad.exe', 'C:\\test.txt')

2. os.call(['C:\\Temp\\a b c\\Notepad.exe', 'C:\\test.txt'])

**3. import subprocess  
subprocess.call(['C:\\Temp\\a b c\\Notepad.exe', 'C:\\test.txt'])**

4. subprocess.call(['C:\\Temp\\a b c\\Notepad.exe', 'C:\\test.txt'])

### 85. Which of the following is the base class for new-style file objects?

1. base

**2. file**

3. fileobject

4. filebase

### 86. What is the difference between the two methods, "foo is None", and "foo == None", for the purposes of determining if the type of an object is None?

**1. "is" always returns True if it compares the same object instance whereas the value returned by "==" is determined by the \_\_eq\_\_() method.**

2. "==" always returns True if it compares the same object instance whereas the value returned by "is" is determined by the \_\_eq\_\_() method.

3. There is no difference between the two, but "==" is preferred by convention.

4. There is no difference between the two, but "is" is preferred by convention.

### 87. Given a program that has different combinations of conditions, programming a custom function for each combined condition can produce an excessive number of functions. Instead, for conditions that should be jointly satisfied, several filters can be nested within each other. Which of the following statements does this correctly?

1. short\_regvals = filter(shortline,(isRegDBVal, lines))

2. short\_regvals = filter(filter(isRegDBVal, lines),shortline)

**3. short\_regvals = filter(shortline, filter(isRegDBVal, lines))**

4. None of these

### 88. Given the two dictionaries 'x' and 'y', which of these codes would not create a final, merged dictionary 'z'? x = {'a':1, 'b': 2} y = {'b':10, 'c': 11} z = {'a': 1, 'c': 11, 'b': 10}

1. z = dict(x.items() + y.items())

2. z = x.copy()  
z.update(y)

3. z = dict(x, \*\*y)

**4. z = x.update(y)**

5. z = dict(list(x.items()) + list(y.items()))

### 89. It is possible to use encoding other than ASCII in Python source files. The best way to do it is to put one more special comment line right after the #! line to define the source file encoding. Which of the following is the correct statement to define encoding?

1. coding: iso-8859-15

2. # -\*- iso-8859-15 -\*-

**3. # -\*- coding: iso-8859-15 -\*-**

4. None of these

### 90. Which of the following code samples will determine the last day of a given month?

1. import datetime  
datetime.date(year, month, day).replace(day=1,month=datetime.date(year, month, day).month+1)-timedelta(days=1)

**2. import calendar  
calendar.monthrange(year, month)[1]**

3. import datetime  
  
dt = datetime.date(year, month, day)  
next\_month = (dt.month % 12) + 1  
end\_date = datetime.date(dt.year, next\_month, 1) - datetime.timedelta(days=1)  
return end\_date

4. import calendar  
calendar.lastday(year, month)

### 91. Which of the following statements is correct for the interpreter when it is in interactive mode?

1. Commands are read from a tty

**2. Signs (">>> ") for primary prompt**

3. Signs ("<<< ") for secondary prompt

### 92. What is the result of the following code: >>> import itertools >>> x = itertools.count(0) >>> x.\_\_class\_\_.\_\_name\_\_

1. Nothing

**2. 'count'**

3. 'Object'

4. 'None'

### 93. The most important element of the email package is the message. The email class provides the classes for messages in Python. Which of the following classes is used for the message?

1. email.Message

**2. email.message.Message**

3. email.messages

4. email.emailmessage