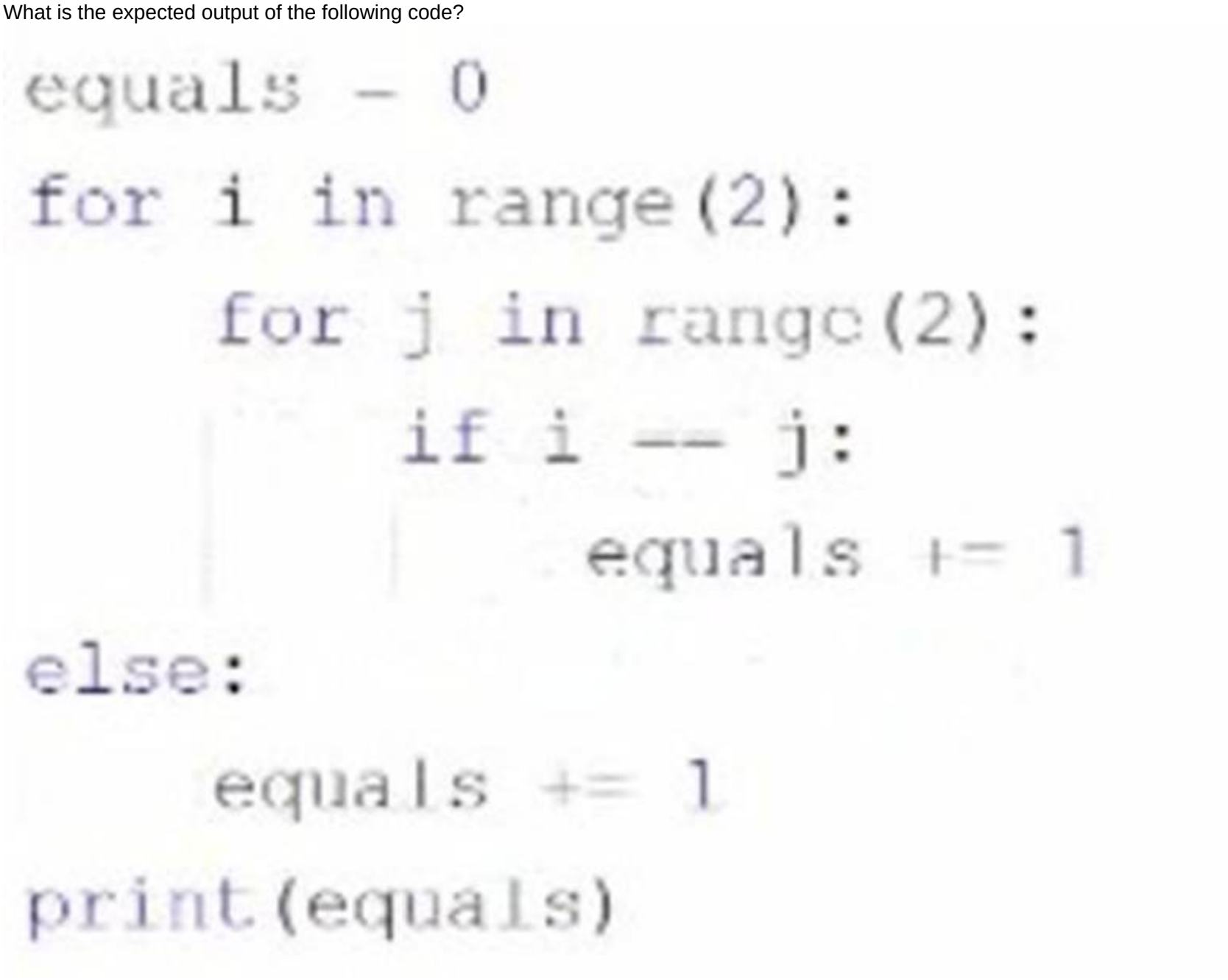
Python-Institute

Exam Questions PCEP-30-02

PCEP - Certified Entry-Level Python Programmer

NEW QUESTION 1
What is the expected output of the following code?



A. The code outputs nothing.
B. 3
C. 1
D. 4

Answer: C

Explanation:
The code snippet that you have sent is checking if two numbers are equal and printing the result. The code is as follows:
num1 = 1 num2 = 2 if num1 == num2: print(4) else: print(1)
The code starts with assigning the values 1 and 2 to the variables ??num1?? and ??num2?? respectively. Then, it enters an if statement that compares the values of ??num1?? and ??num2?? using the equality operator (==). If the values are equal, the code prints 4 to the screen. If the values are not equal, the code prints 1 to the screen.
The expected output of the code is 1, because the values of ??num1?? and ??num2?? are not equal. Therefore, the correct answer is C. 1.
Reference: [Python Institute - Entry-Level Python Programmer Certification]

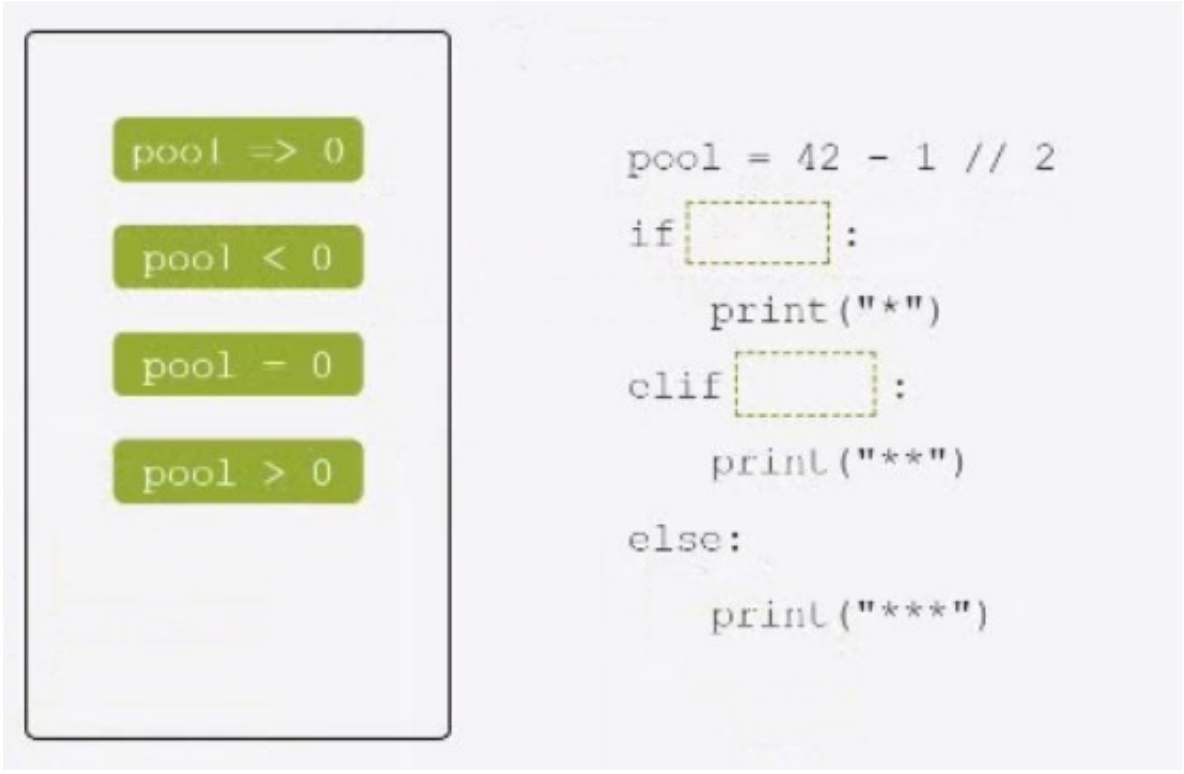
NEW QUESTION 2
Which of the following are the names of Python passing argument styles? (Select two answers.)

A. keyword
B. reference
C. indicatory
D. positional

Answer: AD

Explanation:
Keyword arguments are arguments that are specified by using the name of the parameter, followed by an equal sign and the value of the argument. For example, print (sep='-', end='!') is a function call with keyword arguments. Keyword arguments can be used to pass arguments in any order, and to provide default values for some arguments1.
Positional arguments are arguments that are passed in the same order as the parameters of the function definition. For example, print ('Hello', 'World') is a function call with positional arguments. Positional arguments must be passed before any keyword arguments, and they must match the number and type of the parameters of the function2.
References: 1: 5 Types of Arguments in Python Function Definitions | Built In 2: python - What??s the pythonic way to pass arguments between functions ??

NEW QUESTION 3
DRAG DROP
Drag and drop the conditional expressions to obtain a code which outputs * to the screen. (Note: some code boxes will not be used.)



A. Mastered
B. Not Mastered

Answer: A

Explanation:
One possible way to drag and drop the conditional expressions to obtain a code which outputs * to the screen is:
if pool > 0: print("*")
elif pool < 0: print("**") else: print("***")
This code uses the if, elif, and else keywords to create a conditional statement that checks
the value of the variable pool. Depending on whether the value is greater than, less than, or equal to zero, the code will print a different pattern of asterisks to the screen.
The print function is used to display the output. The code is indented to show the blocks of code that belong to each condition. The code will output * if the value of pool is positive, ** if the value of pool is negative, and *** if the value of pool is zero.
You can find more information about the conditional statements and the print function in Python in the following references:
? [Python If ?? Else]
? [Python Print Function]
? [Python Basic Syntax]

NEW QUESTION 4
What is true about exceptions and debugging? (Select two answers.)

A. A tool that allows you to precisely trace program execution is called a debugger.
B. If some Python code is executed without errors, this proves that there are no errors in it.
C. One try-except block may contain more than one except branch.
D. The default (anonymous) except branch cannot be the last branch in the try-except block.

Answer: AC

Explanation:
Exceptions and debugging are two important concepts in Python programming that are related to handling and preventing errors. Exceptions are errors that occur when the code cannot be executed properly, such as syntax errors, type errors, index errors, etc. Debugging is the process of finding and fixing errors in the code, using various tools and techniques. Some of the facts about exceptions and debugging are:
? A tool that allows you to precisely trace program execution is called a debugger. A debugger is a program that can run another program step by step, inspect the values of variables, set breakpoints, evaluate expressions, etc. A debugger can help you find the source and cause of an error, and test possible solutions. Python has a built-in debugger module called pdb, which can be used from the command line or within the code. There are also other third-party debuggers available for Python, such as PyCharm, Visual Studio Code, etc12
? If some Python code is executed without errors, this does not prove that there are no errors in it. It only means that the code did not encounter any exceptions that would stop the execution. However, the code may still have logical errors, which are errors that cause the code to produce incorrect or unexpected results. For example, if you write a function that is supposed to calculate the area of a circle, but you use the wrong formula, the code may run without errors, but it will give you the wrong answer. Logical errors are harder to detect and debug than syntax or runtime errors, because they do not generate any error messages. You have to test the code with different inputs and outputs, and compare them with the expected results34
? One try-except block may contain more than one except branch. A try-except block is a way of handling exceptions in Python, by using the keywords try and except. The try block contains the code that may raise an exception, and the except block contains the code that will execute if an exception occurs. You can have multiple except blocks for different types of exceptions, or for different actions to take. For example, you can write a try-except block like this:
try: # some code that may raise an exception except ValueError: # handle the ValueError exception except ZeroDivisionError: # handle the ZeroDivisionError exception except: # handle any other exception
This way, you can customize the error handling for different situations, and provide more informative messages or alternative solutions5
? The default (anonymous) except branch can be the last branch in the try-except block. The default except branch is the one that does not specify any exception type, and it will catch any exception that is not handled by the previous except branches. The default except branch can be the last branch in the try-except block, but it cannot be the first or the only branch. For example, you can write a try- except block like this:
try: # some code that may raise an exception except ValueError: # handle the ValueError exception except: # handle any other exception
This is a valid try-except block, and the default except branch will be the last branch. However, you cannot write a try-except block like this:
try: # some code that may raise an exception except: # handle any exception
This is an invalid try-except block, because the default except branch is the only branch, and it will catch all exceptions, even those that are not errors, such as KeyboardInterrupt or SystemExit. This is considered a bad practice, because it may hide or ignore important exceptions that should be handled differently or propagated further. Therefore, you should always specify the exception types that you want to handle, and use the default except branch only as a last resort5
Therefore, the correct answers are A. A tool that allows you to precisely trace program execution is called a debugger. and C. One try-except block may contain

more than one except branch.
Reference: Python Debugger – Python pdb - GeeksforGeeksHow can I see the details of an exception in Python??s debugger?Python Debugging (fixing problems)Python - start interactive debugger when exception would be otherwise thrownPython Try Except [Error Handling and Debugging — Programming with Python for Engineers]

NEW QUESTION 5
Which of the following expressions evaluate to a non-zero result? (Select two answers.)

A. 2 ** 3 / A - 2
B. 4 / 2 * * 3 - 2
C. 1 * * 3 / 4 - 1
D. 1 * 4 // 2 ** 3

Answer: AB

Explanation:
In Python, the ** operator is used for exponentiation, the / operator is used for floating-point division, and the // operator is used for integer division. The order of operations is parentheses, exponentiation, multiplication/division, and addition/subtraction. Therefore, the expressions can be evaluated as follows:
* A. 2 ** 3 / A - 2 = 8 / A - 2 (assuming A is a variable that is not zero or undefined) B. 4 / 2 * * 3 - 2 = 4 / 8 - 2 = 0.5 - 2 = -1.5 C. 1 * * 3 / 4 - 1 = 1 / 4 - 1 = 0.25 - 1 = -0.75 D. 1 * 4 // 2 ** 3 = 4 // 8 = 0
Only expressions A and B evaluate to non-zero results.
Reference: [Python Institute - Entry-Level Python Programmer Certification]

NEW QUESTION 6
Assuming that the following assignment has been successfully executed:

```
the_list = ["1", 1, 1.]
```

Which of the following expressions evaluate to True? (Select two expressions.)

A. the_List.index {"1"} in the_list
B. 1.1 in the_list |1:3 |
C. len (the list [0:2]} <3
D. the_lis
E. index {'1'} -- 0

Answer: CD

Explanation:
The code snippet that you have sent is assigning a list of four values to a variable called ??the_list??. The code is as follows:
the_list = [??1??, 1, 1, 1]
The code creates a list object that contains the values ??1??, 1, 1, and 1, and assigns it to the variable ??the_list??. The list can be accessed by using the variable name or by using the index of the values. The index starts from 0 for the first value and goes up to the length of the list minus one for the last value. The index can also be negative, in which case it counts from the end of the list. For example, the_list[0] returns ??1??, and the_list[-1] returns 1.
The expressions that you have given are trying to evaluate some conditions on the list and return a boolean value, either True or False. Some of them are valid, and some of them are invalid and will raise an exception. An exception is an error that occurs when the code cannot be executed properly. The expressions are as follows:
* A. the_List.index {??1??} in the_list: This expression is trying to check if the index of the value ??1?? in the list is also a value in the list. However, this expression is invalid, because it uses curly brackets instead of parentheses to call the index method. The index method is used to return the first occurrence of a value in a list. For example, the_list.index(??1??) returns 0, because ??1?? is the first value in the list. However, the_list.index {??1??} will raise a SyntaxError exception and output nothing.
* B. 1.1 in the_list |1:3 |: This expression is trying to check if the value 1.1 is present in a sublist of the list. However, this expression is invalid, because it uses a vertical bar instead of a colon to specify the start and end index of the sublist. The sublist is obtained by using the slicing operation, which uses square brackets and a colon to get a part of the list. For example, the_list[1:3] returns [1, 1], which is the sublist of the list from the index 1 to the index 3, excluding the end index. However, the_list |1:3 | will raise a SyntaxError exception and output nothing.
* C. len (the list [0:2]} <3: This expression is trying to check if the length of a sublist of the list is less than 3. This expression is valid, because it uses the len function and the slicing operation correctly. The len function is used to return the number of values in a list or a sublist. For example, len(the_list) returns 4, because the list has four values. The slicing operation is used to get a part of the list by using square brackets and a colon. For example, the_list[0:2] returns [??1??, 1], which is the sublist of the list from the index 0 to the index 2, excluding the end index. The expression len (the list [0:2]} <3 returns True, because the length of the sublist [??1??, 1] is 2, which is less than 3.
* D. the_list. index {??1??} – 0: This expression is trying to check if the index of the value ??1?? in the list is equal to 0. This expression is valid, because it uses the index method and the equality operator correctly. The index method is used to return the first occurrence of a value in a list. For example, the_list.index(??1??) returns 0, because ??1?? is the first value in the list. The equality operator is used to compare two values and return True if they are equal, or False if they are not. For example, 0 == 0 returns True, and 0 == 1 returns False. The expression the_list. index {??1??} – 0 returns True, because the index of ??1?? in the list is 0, and 0 is equal to 0.
Therefore, the correct answers are C. len (the list [0:2]} <3 and D. the_list. index {??1??} – 0. Reference: Python List Methods - W3Schools5. Data Structures — Python 3.11.5 documentationList methods in Python - GeeksforGeeks

NEW QUESTION 7
DRAG DROP
Arrange the code boxes in the correct positions to form a conditional instruction which guarantees that a certain statement is executed when the speed variable is less than 50.0.

A. Mastered
B. Not Mastered

Answer: A

Explanation:
One possible way to arrange the code boxes in the correct positions to form a conditional instruction which guarantees that a certain statement is executed when the speed variable is less than 50.0 is: if speed < 50.0: print("The speed is low.")
This code uses the if keyword to create a conditional statement that checks the value of the variable speed. If the value is less than 50.0, then the code will print ??The speed is low.?? to the screen. The print function is used to display the output. The code is indented to show the block of code that belongs to the if condition.
You can find more information about the if statement and the print function in Python in the following references:
? Python If ?? Else
? Python Print Function

NEW QUESTION 8
Which of the following functions can be invoked with two arguments?
A)

```
def mu(Nonc):
    pass
```

B)

```
def iota(level, size = 0):
    pass
```

C)

```
def kappa(level):
    pass
```

D)

A. Option A
B. Option B
C. Option C
D. Option D

Answer: B

Explanation:
The code snippets that you have sent are defining four different functions in Python. A function is a block of code that performs a specific task and can be reused in the program. A function can take zero or more arguments, which are values that are passed to the function when it is called. A function can also return a value or None, which is the default return value in Python.
To define a function in Python, you use the def keyword, followed by the name of the function and parentheses. Inside the parentheses, you can specify the names of the parameters that the function will accept. After the parentheses, you use a colon and then indent the code block that contains the statements of the function. For example:
def function_name(parameter1, parameter2): # statements of the function return value To call a function in Python, you use the name of the function followed by parentheses.
Inside the parentheses, you can pass the values for the arguments that the function expects. The number and order of the arguments must match the number and order of the parameters in the function definition, unless you use keyword arguments or default values. For example:
function_name(argument1, argument2)
The code snippets that you have sent are as follows:
A) def my_function(): print(??Hello??)
B) def my_function(a, b): return a + b
C) def my_function(a, b, c): return a * b * c
D) def my_function(a, b=0): return a - b
The question is asking which of these functions can be invoked with two arguments. This means that the function must have two parameters in its definition, or one parameter with a default value and one without. The default value is a value that is assigned to a parameter if no argument is given for it when the function is called. For example, in option D, the parameter b has a default value of 0, so the function can be called with one or two arguments.
The only option that meets this criterion is option B. The function in option B has two parameters, a and b, and returns the sum of them. This function can be invoked with two arguments, such as my_function(2, 3), which will return 5.
The other options cannot be invoked with two arguments. Option A has no parameters, so it can only be called with no arguments, such as my_function(), which will print ??Hello??. Option C has three parameters, a, b, and c, and returns the product of them. This function can only be called with three arguments, such as my_function(2, 3, 4), which will return 24. Option D has one parameter with a default value, b, and one without, a, and returns the difference of them. This function can be called with one or two arguments, such as my_function(2) or my_function(2, 3), which will return 2 or -1, respectively.
Therefore, the correct answer is B. Option B.

NEW QUESTION 9
What is true about tuples? (Select two answers.)

A. Tuples are immutable, which means that their contents cannot be changed during their lifetime.
B. The len { } function cannot be applied to tuples.
C. An empty tuple is written as { } .
D. Tuples can be indexed and sliced like lists.

Answer: AD

Explanation:
Tuples are one of the built-in data types in Python that are used to store collections of data. Tuples have some characteristics that distinguish them from other data types, such as lists, sets, and dictionaries. Some of these characteristics are:
? Tuples are immutable, which means that their contents cannot be changed during their lifetime. Once a tuple is created, it cannot be modified, added, or removed. This makes tuples more stable and reliable than mutable data types. However, this also means that tuples are less flexible and dynamic than mutable data types. For example, if you want to change an element in a tuple, you have to create a new tuple with the modified element and assign it to the same variable12
? Tuples are ordered, which means that the items in a tuple have a defined order and can be accessed by using their index. The index of a tuple starts from 0 for the first item and goes up to the length of the tuple minus one for the last item. The index can also be negative, in which case it counts from the end of the tuple. For example, if you have a tuple t = ("a", "b", "c"), then t[0] returns "a", and t[- 1] returns "c"12
? Tuples can be indexed and sliced like lists, which means that you can get a single item or a sublist of a tuple by using square brackets and specifying the start and end index. For example, if you have a tuple t = ("a", "b", "c", "d", "e"),
then t[2] returns "c", and t[1:4] returns ("b", "c", "d"). Slicing does not raise any exception, even if the start or end index is out of range. It will just return an empty tuple or the closest possible sublist12
? Tuples can contain any data type, such as strings, numbers, booleans, lists, sets,
dictionaries, or even other tuples. Tuples can also have duplicate values, which means that the same item can appear more than once in a tuple. For example, you can have a tuple t = (1, 2, 3, 1, 2), which contains two 1s and two 2s12
? Tuples are written with round brackets, which means that you have to enclose the
items in a tuple with parentheses. For example, you can create a tuple t = ("a", "b", "c") by using round brackets. However, you can also create a tuple without using round brackets, by just separating the items with commas. For example, you can create the same tuple t = "a", "b", "c" by using commas. This is called tuple packing, and it allows you to assign multiple values to a single variable12
? The len() function can be applied to tuples, which means that you can get the
number of items in a tuple by using the len() function. For example, if you have a tuple t = ("a", "b", "c"), then len(t) returns 312

? An empty tuple is written as (), which means that you have to use an empty pair of
parentheses to create a tuple with no items. For example, you can create an empty tuple t = () by using empty parentheses. However, if you want to create a tuple with only one item, you have to add a comma after the item, otherwise Python will not recognize it as a tuple. For example, you can create a tuple with one item t = ("a",) by using a comma12
Therefore, the correct answers are A. Tuples are immutable, which means that their contents cannot be changed during their lifetime. and D. Tuples can be indexed and sliced like lists.
Reference: Python Tuples - W3SchoolsTuples in Python - GeeksforGeeks

NEW QUESTION 10
.....

Thank You for Trying Our Product

We offer two products:

1st - We have Practice Tests Software with Actual Exam Questions

2nd - Questons and Answers in PDF Format

PCEP-30-02 Practice Exam Features:

* PCEP-30-02 Questions and Answers Updated Frequently

* PCEP-30-02 Practice Questions Verified by Expert Senior Certified Staff

* PCEP-30-02 Most Realistic Questions that Guarantee you a Pass on Your FirstTry

* PCEP-30-02 Practice Test Questions in Multiple Choice Formats and Updatesfor 1 Year