Python-Institute

Exam Questions PCEP-30-02

PCEP - Certified Entry-Level Python Programmer

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

```
rates = (1.2, 1.4, 1.0)
new = rates[3:]
for rate in rates[-2:]:
    new += (rate,)
print(len(new))
```

A. 5
B. 2
C. 1
D. The code will cause an unhandled

Answer: D

Explanation:
The code snippet that you have sent is trying to use a list comprehension to create a new list from an existing list. The code is as follows:
my_list = [1, 2, 3, 4, 5] new_list = [x for x in my_list if x > 5]
The code starts with creating a list called ??my_list?? that contains the numbers 1, 2, 3, 4, and 5. Then, it tries to create a new list called ??new_list?? by using a list comprehension. A list comprehension is a concise way of creating a new list from an existing list by applying some expression or condition to each element.
The syntax of a list comprehension is:
new_list = [expression for element in old_list if condition]
The expression is the value that will be added to the new list, which can be the same as the element or a modified version of it. The element is the variable that takes each value from the old list. The condition is an optional filter that determines which elements will be included in the new list. For example, the following list comprehension creates a new list that contains the squares of the even numbers from the old list:
old_list = [1, 2, 3, 4, 5, 6] new_list = [x ** 2 for x in old_list if x % 2 == 0]
new_list = [4, 16, 36]The code that you have sent is trying to create a new list that contains the elements from the old list that are greater than 5. However, there is a problem with this code. The problem is that none of the elements in the old list are greater than 5, so the condition is always false. This means that the new list will be empty, and the expression will never be evaluated. However, the expression is not valid, because it uses the variable x without defining it. This will cause a NameError exception, which is an error that occurs when a variable name is not found in the current scope. The code does not handle the exception, and therefore it will terminate with an error message.
The expected result of the code is an unhandled exception, because the code tries to use an undefined variable in an expression that is never executed.
Therefore, the correct answer is D. The code will cause an unhandled exception.
Reference: Python - List Comprehension - W3SchoolsPython - List Comprehension -
GeeksforGeeksPython Exceptions: An Introduction – Real Python

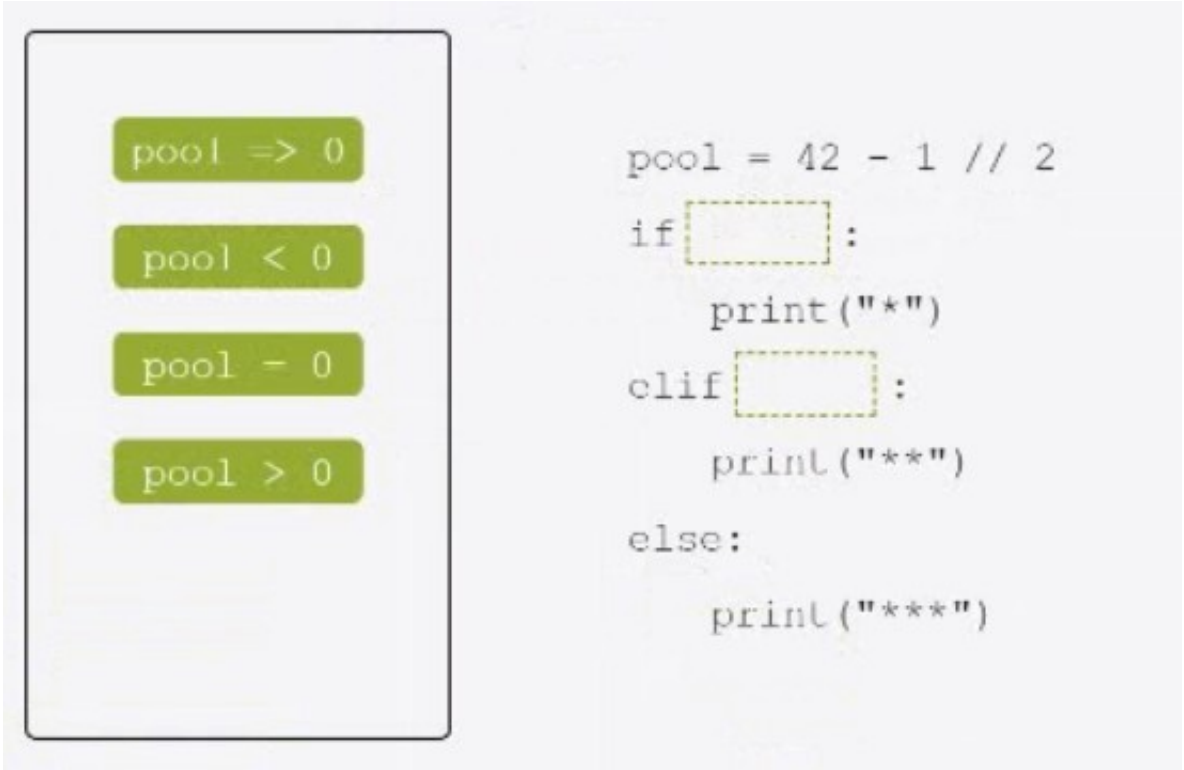
NEW QUESTION 2
Which of the following are the names of Python passing argument styles? (Select two answers.)

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.)

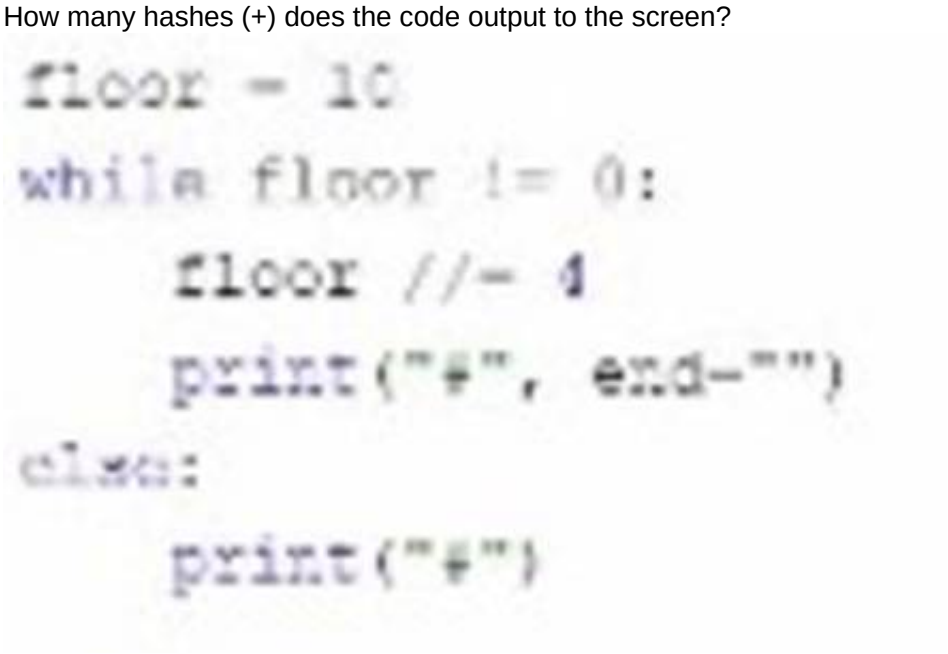


A. Mastered
B. Not Mastered

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
How many hashes (+) does the code output to the screen?



A. one
B. zero (the code outputs nothing)
C. five
D. three

Answer: C

Explanation:
The code snippet that you have sent is a loop that checks if a variable ??floor?? is less than or equal to 0 and prints a string accordingly. The code is as follows:
floor = 5 while floor > 0: print(??+??) floor = floor - 1
The code starts with assigning the value 5 to the variable ??floor??. Then, it enters a while loop that repeats as long as the condition ??floor > 0?? is true. Inside the loop, the code prints a ??+?? symbol to the screen, and then subtracts 1 from the value of ??floor??. The loop ends when ??floor?? becomes 0 or negative, and the code exits.
The code outputs five ??+?? symbols to the screen, one for each iteration of the loop. Therefore, the correct answer is C. five.
Reference: [Python Institute - Entry-Level Python Programmer Certification]

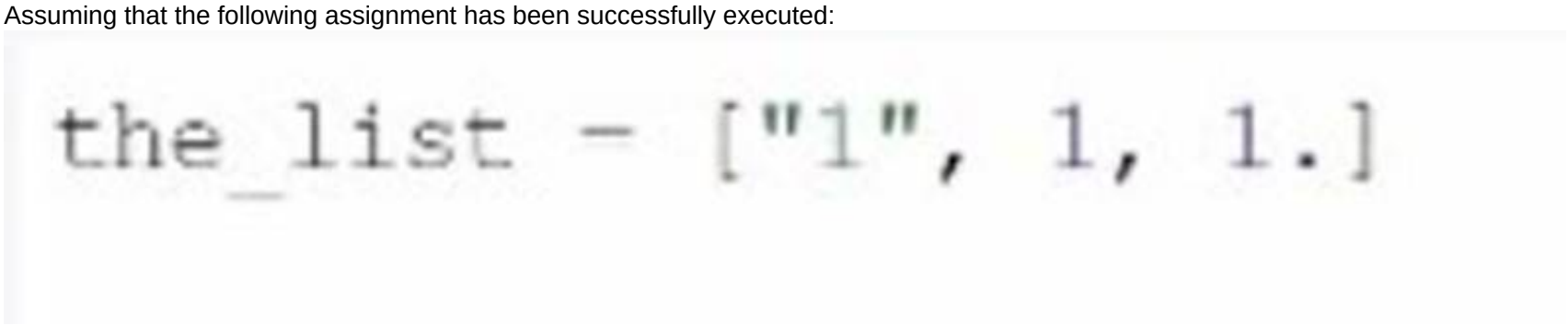
NEW QUESTION 5
Which of the following expressions evaluate to a non-zero result? (Select two answers.)

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:



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
What is the expected output of the following code?

```
menu = {"pizza": 2.39, "pasta": 1.99, "folpetti": 3.99}

for value in menu:
    print(str(value)[0], end="")
```

A. The code is erroneous and cannot be run.
B. ppt
C. 213
D. pizzapastafolpetti

Answer: B

Explanation:
The code snippet that you have sent is using the slicing operation to get parts of a string and concatenate them together. The code is as follows:
pizza = ??pizza?? pasta = ??pasta?? folpetti = ??folpetti?? print(pizza[0] + pasta[0] + folpetti[0])
The code starts with assigning the strings ??pizza??, ??pasta??, and ??folpetti?? to the variables pizza, pasta, and folpetti respectively. Then, it uses the print function to display the result of concatenating the first characters of each string. The first character of a string can be accessed by using the index 0 inside square brackets. For example, pizza[0] returns ??p??. The concatenation operation is used to join two or more strings together by using the + operator. For example, ??a?? + ??b?? returns ??ab??. The code prints the result of pizza[0] + pasta[0] + folpetti[0], which is ??p?? + ??p?? + ??f??, which is ??ppt??.
The expected output of the code is ppt, because the code prints the first characters of each string. Therefore, the correct answer is B. ppt.
Reference: Python String Slicing - W3SchoolsPython String Concatenation - W3Schools

NEW QUESTION 8
What is the expected output of the following code?

```
counter = 84 // 2
if counter < 0:
    print("*")
elif counter >= 42:
    print("**")
else:
    print("***")
```

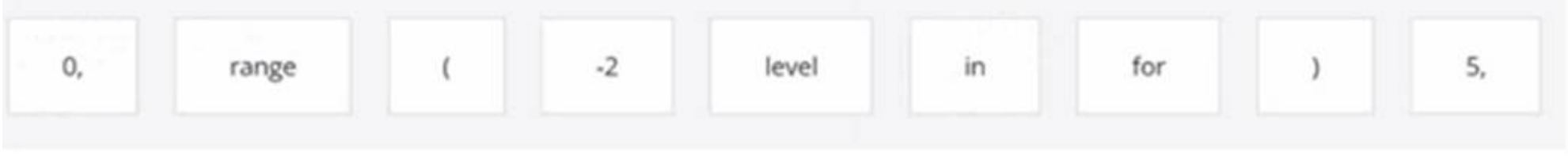
A. The code produces no output.
B. * * *
C. * *
D. *

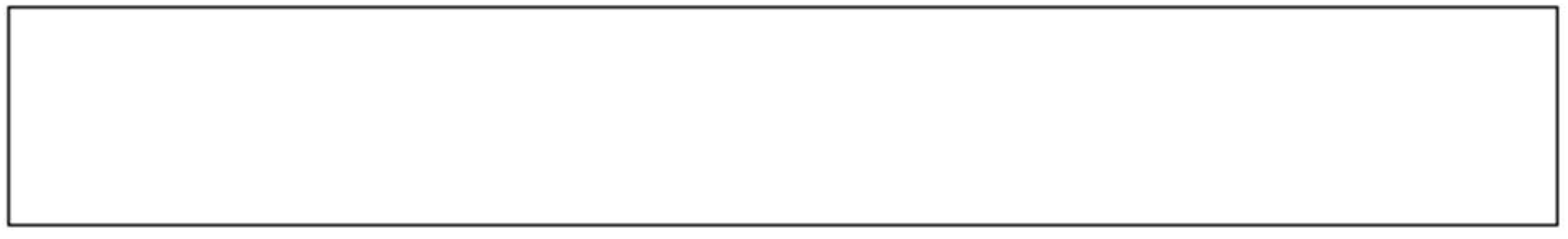
Answer: C

Explanation:
The code snippet that you have sent is a conditional statement that checks if a variable ??counter?? is less than 0, greater than or equal to 42, or neither. The code is as follows: if counter < 0: print(????) elif counter >= 42: print(????) else: print(????)
The code starts with checking if the value of ??counter?? is less than 0. If yes, it prints a single asterisk () to the screen and exits the statement. If no, it checks if the value of ??counter?? is greater than or equal to 42. If yes, it prints three asterisks () to the screen and exits the statement. If no, it prints two asterisks () to the screen and exits the statement.

The expected output of the code depends on the value of ??counter??. If the value of ??counter?? is 10, as shown in the image, the code will print two asterisks (**) to the screen, because 10 is neither less than 0 nor greater than or equal to 42. Therefore, the correct answer is C.
* *
Reference: [Python Institute - Entry-Level Python Programmer Certification]

NEW QUESTION 9
DRAG DROP
Arrange the code boxes in the correct positions in order to obtain a loop which executes its body with the level variable going through values 5, 1, and 1 (in the same order).

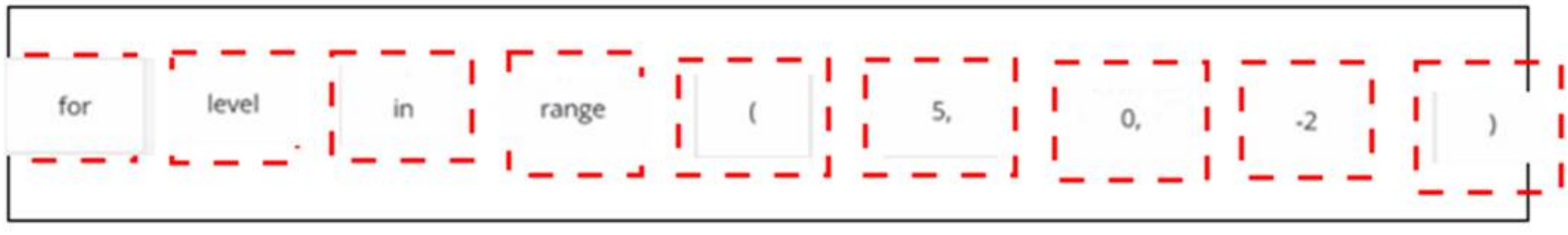




A. Mastered
B. Not Mastered

Answer: A

Explanation:



NEW QUESTION 10
Assuming that the following assignment has been successfully executed: My_list – [1, 1, 2, 3]
Select the expressions which will not raise any exception. (Select two expressions.)

A. my_list[-10]
B. my_list|my_Li1st | 3| I
C. my list [6]
D. my_List- [0:1]

Answer: BD

Explanation:
The code snippet that you have sent is assigning a list of four numbers to a variable called ??my_list??. The code is as follows:
my_list = [1, 1, 2, 3]
The code creates a list object that contains the elements 1, 1, 2, and 3, and assigns it to the variable ??my_list??. The list can be accessed by using the variable name or by using the index of the elements. The index starts from 0 for the first element and goes up to the length of the list minus one for the last element. The index can also be negative, in which case it counts from the end of the list. For example, my_list[0] returns 1, and my_list[-1] returns 3.
The code also allows some operations on the list, such as slicing, concatenation, repetition, and membership. Slicing is used to get a sublist of the original list by specifying the start and end index. For example, my_list[1:3] returns [1, 2]. Concatenation is used to join two lists together by using the + operator. For example, my_list + [4, 5] returns [1, 1, 2, 3, 4, 5]. Repetition is used to create a new list by repeating the original list a number of times by using the * operator. For example, my_list * 2 returns [1, 1, 2, 3, 1, 1, 2, 3]. Membership is used to check if an element is present in the list by using the in operator. For example, 2 in my_list returns True, and 4 in my_list returns False.
The expressions that you have given are trying to access or manipulate the list in different ways. 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. my_list[-10]: This expression is trying to access the element at the index -10 of the list. However, the list only has four elements, so the index -10 is out of range. This will raise an IndexError exception and output nothing.
* B. my_list|my_Li1st | 3| I: This expression is trying to perform a bitwise OR operation on the list and some other operands. The bitwise OR operation is used to

compare the binary representation of two numbers and return a new number that has a 1 in each bit position where either number has a 1. For example, 3 | 1 returns 3, because 3 in binary is 11 and 1 in binary is 01, and 11 | 01 is 11. However, the bitwise OR operation cannot be applied to a list, because a list is not a number. This will raise a TypeError exception and output nothing.
* C. my list [6]: This expression is trying to access the element at the index 6 of the list. However, the list only has four elements, so the index 6 is out of range. This will raise an IndexError exception and output nothing.
* D. my_List- [0:1]: This expression is trying to perform a subtraction operation on the list and a sublist. The subtraction operation is used to subtract one number from another and return the difference. For example, 3 - 1 returns 2. However, the subtraction operation cannot be applied to a list, because a list is not a number. This will raise a TypeError exception and output nothing.
Only two expressions will not raise any exception. They are:
* B. my_list|my_Li1st | 3| I: This expression is not a valid Python code, but it is not an expression that tries to access or manipulate the list. It is just a string of characters that has no meaning. Therefore, it will not raise any exception, but it will also not output anything.
* D. my_List- [0:1]: This expression is a valid Python code that uses the slicing operation to get a sublist of the list. The slicing operation does not raise any exception, even if the start or end index is out of range. It will just return an empty list or the closest possible sublist.
For example, my_list[0:10] returns [1, 1, 2, 3], and my_list[10:20] returns []. The expression my_List- [0:1] returns the sublist of the list from the index 0 to the index 1, excluding the end index. Therefore, it returns [1]. This expression will not raise any exception, and it will output [1].
Therefore, the correct answers are B. my_list|my_Li1st | 3| I and D. my_List- [0:1]. Reference: [Python Institute - Entry-Level Python Programmer Certification]

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