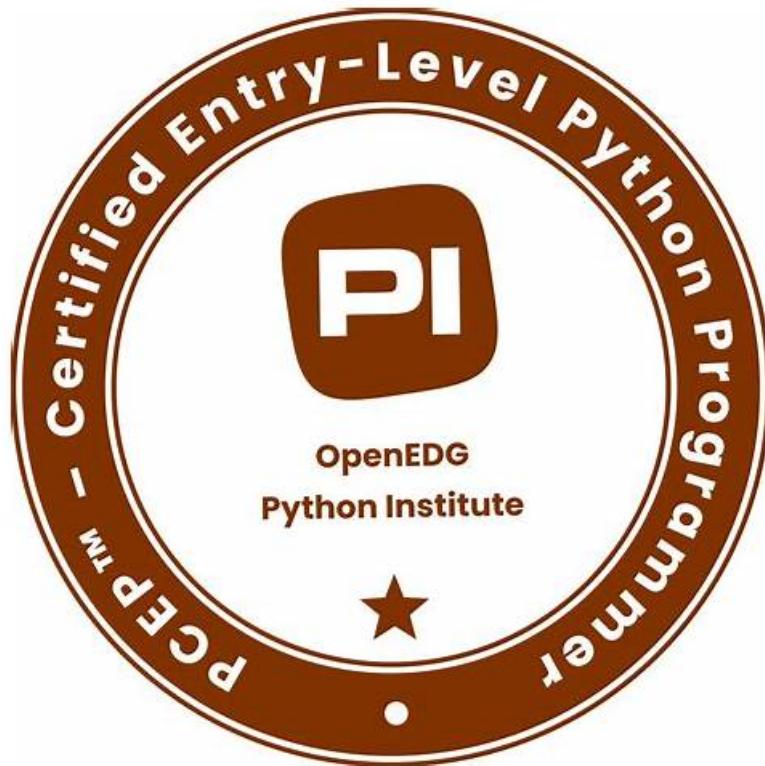


Pass Guaranteed Python Institute PCEP-30-02 - PCEP - Certified Entry-Level Python Programmer Updated Valid Test Objectives



What's more, part of that Test4Sure PCEP-30-02 dumps now are free: <https://drive.google.com/open?id=1B-bc8p00jxKg8Mu8m6RslHkgsKgrjRH->

To fulfill our dream of helping our users get the PCEP-30-02 certification more efficiently, we are online to serve our customers 24 hours a day and 7 days a week. Therefore, whenever you have problems in studying our PCEP-30-02 test training, we are here for you. You can contact with us through e-mail or just send to our message online. And unlike many other customer service staff who have bad temper, our staff are gentle and patient enough for any of your problems in practicing our PCEP-30-02 study torrent. In addition, we have professional personnel to give you remote assistance on PCEP-30-02 exam questions.

Python Institute PCEP-30-02 Exam Syllabus Topics:

Topic	Details
Topic 1	<ul style="list-style-type: none">Control Flow: This section covers conditional statements such as if, if-else, if-elif, if-elif-else
Topic 2	<ul style="list-style-type: none">parameters, arguments, and scopes. It also covers Recursion, Exception hierarchy, Exception handling, etc.
Topic 3	<ul style="list-style-type: none">Functions and Exceptions: This part of the exam covers the definition of function and invocation
Topic 4	<ul style="list-style-type: none">Computer Programming Fundamentals: This section of the exam covers fundamental concepts such as interpreters, compilers, syntax, and semantics. It covers Python basics: keywords, instructions, indentation, comments in addition to Booleans, integers, floats, strings, and Variables, and naming conventions. Finally, it covers arithmetic, string, assignment, bitwise, Boolean, relational, and Inputoutput operations.

PCEP-30-02 Examcollection, PCEP-30-02 Exam Simulations

In order to save a lot of unnecessary trouble to users, we have completed our PCEP - Certified Entry-Level Python Programmer study questions research and development of online learning platform, users do not need to download and install, only need your digital devices have a browser, can be done online operation of the PCEP-30-02 test guide. This kind of learning method is very convenient for the user, especially in the time of our fast pace to get Python Institute certification. In addition, our test data is completely free of user's computer memory, will only consume a small amount of running memory when the user is using our product. At the same time, as long as the user ensures that the network is stable when using our PCEP-30-02 Training Materials, all the operations of the learning material of can be applied perfectly.

Python Institute PCEP - Certified Entry-Level Python Programmer Sample Questions (Q34-Q39):

NEW QUESTION # 34

What happens when the user runs the following code?

□

- A. The program outputs three asterisks(***) to the screen.
- B. The program enters an infinite loop.
- C. The program outputs five asterisks (*****) to the screen.
- D. The program outputs one asterisk (*) to the screen.

Answer: D

NEW QUESTION # 35

Insert the code boxes in the correct positions in order to build a line of code which asks the user for a float value and assigns it to the mass variable.

(Note: some code boxes will not be used.)

□

Answer:

Explanation:

□

Explanation:

□

One possible way to insert the code boxes in the correct positions in order to build a line of code that asks the user for a float value and assigns it to the mass variable is:

mass = float(input("Enter the mass: "))

This line of code uses the input function to prompt the user for a string value, and then uses the float function to convert that string value into a floating-point number. The result is then assigned to the variable mass.

You can find more information about the input and float functions in Python in the following references:

* [Python input() Function]

* [Python float() Function]

NEW QUESTION # 36

What is the expected result of the following code?

□

- A. 0
- B. 1
- C. 2
- D. The code will cause an unhandled

Answer: D

Explanation:

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.

NEW QUESTION # 37

What is the expected result of the following code?

- A. 0
- B. 1
- C. 2
- 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 # 38

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- [0:1]`
- B. `my list [6]`
- C. `my_list|my_Lilst | 3| I`
- D. `my_list[-10]`

Answer: A,C

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_Lilst | 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_Lilst | 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_Lilst | 3| I` and D. `my_List- [0:1]`.

Reference: [Python Institute - Entry-Level Python Programmer Certification]

NEW QUESTION # 39

.....

To eliminate the chances of mistakes and prepare well for exams you must use PCEP-30-02 practice test software. There are two types of PCEP - Certified Entry-Level Python Programmer PCEP-30-02 practice test software: You can install Python Institute PCEP-30-02 practice test software on all window-based PCs. On the other hand, a web-based PCEP - Certified Entry-Level Python Programmer Networking Solutions PCEP-30-02 practice test can be used without the installation of any software. Practicing with these PCEP-30-02 practice exams software seems like you are taking a Real PCEP-30-02 Exam. This software allows you to take multiple Python Institute PCEP-30-02 exam attempts. At the end of each PCEP - Certified Entry-Level Python Programmer PCEP-30-02 exam attempt, you can check your progress. These Python Institute PCEP-30-02 practice tests assist you to know how to manage your time and complete the PCEP - Certified Entry-Level Python Programmer PCEP-30-02 exam within the specified time limit. Thus, Using these PCEP-30-02 practice tests software will be beneficial if you want to achieve the highest score

in the exam.

PCEP-30-02 Examcollection: <https://www.test4sure.com/PCEP-30-02-pass4sure-vce.html>

P.S. Free 2026 Python Institute PCEP-30-02 dumps are available on Google Drive shared by Test4Sure: <https://drive.google.com/open?id=1B-bc8p00jxKg8Mu8m6RsIHkgsKgrjRH->