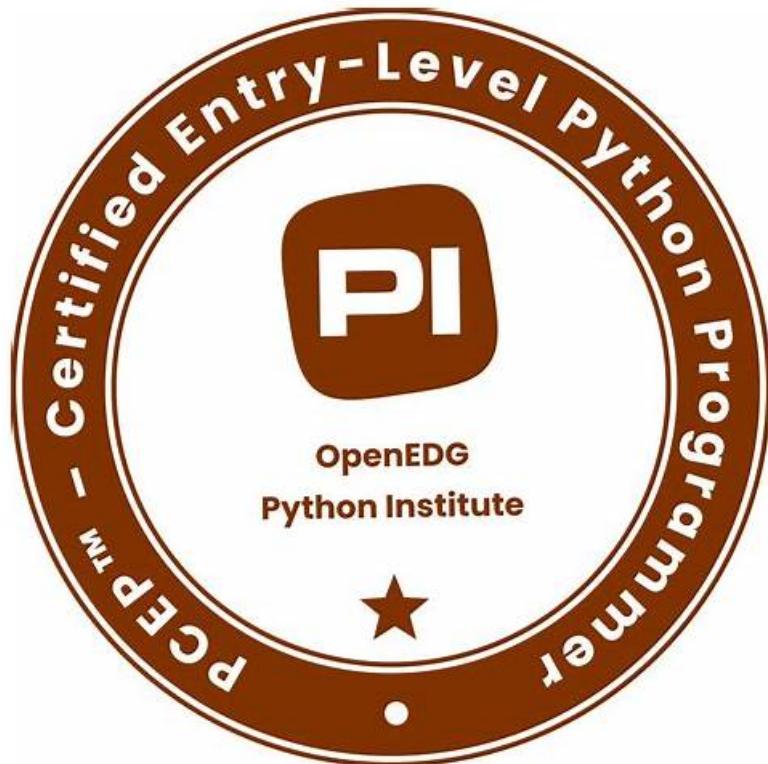


Python Institute - PCEP-30-02 - Valid PCEP - Certified Entry-Level Python Programmer 100% Accuracy



BONUS!!! Download part of Prep4cram PCEP-30-02 dumps for free: <https://drive.google.com/open?id=1n79V0aOSIKLzv5TxGW3XhgQUF-dL9NZe>

In the past ten years, our company has never stopped improving the PCEP-30-02 exam cram. For a long time, we have invested much money to perfect our products. At the same time, we have introduced the most advanced technology and researchers to perfect our PCEP-30-02 exam questions. At present, the overall strength of our company is much stronger than before. We are the leader in the market and master the most advanced technology. In fact, our PCEP-30-02 Test Guide has occupied large market shares because of our consistent renovating. We have built a powerful research center and owned a strong team. Up to now, we have got a lot of patents about the PCEP-30-02 test guide. In the future, we will continuously invest more money on researching.

If you are a workman and you want to pass PCEP-30-02 exam quickly, Prep4cram will be your best choice. PCEP-30-02 dumps and answers from our Prep4cram site are all created by the IT talents with more than 10-year experience in IT certification. It can not only save your time, but also help you pass the PCEP-30-02 Exam easily.

>> PCEP-30-02 100% Accuracy <<

Real PCEP-30-02 Torrent & PCEP-30-02 Latest Learning Material

Prep4cram has many PCEP - Certified Entry-Level Python Programmer (PCEP-30-02) practice questions that reflect the pattern of the real PCEP - Certified Entry-Level Python Programmer (PCEP-30-02) exam. Prep4cram allows you to create a PCEP - Certified Entry-Level Python Programmer (PCEP-30-02) exam dumps according to your preparation. It is easy to create the Python Institute PCEP-30-02 practice questions by following just a few simple steps. Our PCEP - Certified Entry-Level Python Programmer (PCEP-30-02) exam dumps are customizable based on the time and type of questions. You have the option to change the topic and set the time according to the actual PCEP - Certified Entry-Level Python Programmer (PCEP-30-02) exam.

Python Institute PCEP-30-02 Exam Syllabus Topics:

Topic	Details

Topic 1	<ul style="list-style-type: none"> parameters, arguments, and scopes. It also covers Recursion, Exception hierarchy, Exception handling, etc.
Topic 2	<ul style="list-style-type: none"> Control Flow: This section covers conditional statements such as if, if-else, if-elif, if-elif-else
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"> Loops: while, for, range(), loops control, and nesting of loops.

Python Institute PCEP - Certified Entry-Level Python Programmer Sample Questions (Q10-Q15):

NEW QUESTION # 10

What is the expected result of the following code?

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

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 # 11

Which of the following functions can be invoked with two arguments?

- A.
- B.
- C.
- 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 # 12

A program written in a high-level programming language is called:

- A. a binary code
- B. machine code
- C. the ASCII I code
- D. a source code

Answer: D

NEW QUESTION # 13

What is the expected output of the following code?

□

- A. 0
- B. ('Ampere*'. '2021', 'False')
- C. 1
- D. The code raises an unhandled exception.

Answer: C

NEW QUESTION # 14

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

□

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

Answer: B

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 # 15

• • • • •

If you study with our PCEP-30-02 exam questions, you will have a 99% chance to pass the exam. Of course, you don't have to buy any other study materials. Our PCEP-30-02 exam questions can satisfy all your learning needs. During this time, you must really be learning. If you just put PCEP-30-02 Real Exam in front of them and didn't look at them, then we have no way. Our PCEP-30-02 exam questions want to work with you to help you achieve your dreams.

Real PCEP-30-02 Torrent: https://www.prep4cram.com/PCEP-30-02_exam-questions.html

DOWNLOAD the newest Prep4cram PCEP-30-02 PDF dumps from Cloud Storage for free: <https://drive.google.com/open?id=1n79V0aOSIKLzv5TxGW3XhgOUF-dL9NZe>