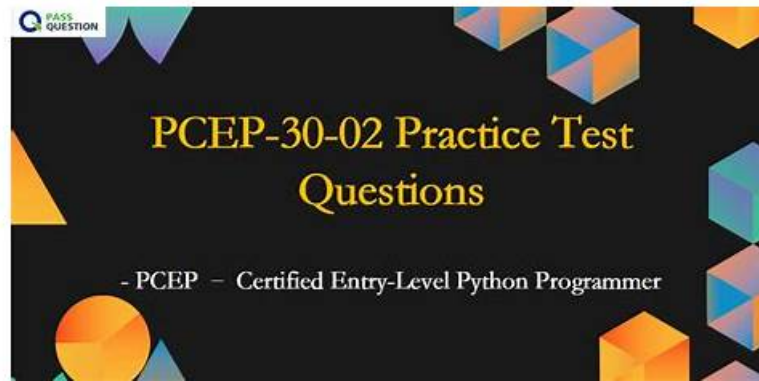


최신버전PCEP-30-02유효한공부문제시험덤프문제



Itexamdump PCEP-30-02 최신 PDF 버전 시험 문제집을 무료로 Google Drive에서 다운로드하세요:
https://drive.google.com/open?id=1OAdX--Gs8eus-W-mXhxMM8YDWL5k_O3

우리는 여러분이 시험패스는 물론 또 일년무료 업데이트서비스를 제공합니다.만약 시험에서 실패했다면 우리는 덤프비용전액 환불을 약속 드립니다.하지만 이런 일은 없을 것입니다.우리는 우리덤프로 100%시험패스에 자신이 있습니다. 여러분은 먼저 우리 Itexamdump사이트에서 제공되는Python Institute인증PCEP-30-02시험덤프의 일부분인 데모 즉 문제와 답을 다운받으셔서 체험해보실 수 있습니다.

Python Institute PCEP-30-02 시험요강:

주제	소개
주제 1	Loops: while, for, range(), loops control, and nesting of loops.
주제 2	Functions and Exceptions: This part of the exam covers the definition of function and invocation
주제 3	Control Flow: This section covers conditional statements such as if, if-else, if-elif, if-elif-else
주제 4	- 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 Input - output operations.
주제 5	parameters, arguments, and scopes. It also covers Recursion, Exception hierarchy, Exception handling, etc.

>> PCEP-30-02유효한 공부문제 <<

Python Institute PCEP-30-02인증덤프문제 - PCEP-30-02최신버전 시험덤프문제

한번에Python Institute인증PCEP-30-02시험을 패스하고 싶으시다면 완전 페펙트한 준비가 필요합니다. 완벽한 관련 지식터득은 물론입니다. 우리Itexamdump의 자료들은 여러분의 이런 시험준비에 많은 도움이 될 것입니다.

최신 Python Institute PCEP PCEP-30-02 무료샘플문제 (Q18-Q23):

질문 # 18

Drag and drop the literals to match their data type names.

□

정답:

설명:

□ Explanation:

□

질문 # 19

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

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

정답: C,D

설명:

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 - GeeksforGeeks
How 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 thrown Python Try Except [Error Handling and Debugging - Programming with Python for Engineers]

질문 # 20

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).

□

정답:

설명:

□

질문 # 21

What is the expected output of the following code?

□

- A. 0
- B. 1
- C. The code raises an exception and outputs nothing.
- D. 2

정답: C

설명:

Explanation

The code snippet that you have sent is trying to print the combined length of two lists, "collection" and "duplicate". The code is as follows:

```
collection = [] collection.append(1) collection.insert(0, 2) duplicate = collection duplicate.append(3) print(len(collection) +
```

len(duplicate)) The code starts with creating an empty list called "collection" and appending the number 1 to it. The list now contains [1]. Then, the code inserts the number 2 at the beginning of the list. The list now contains [2, 1].

Then, the code creates a new list called "duplicate" and assigns it the value of "collection". However, this does not create a copy of the list, but rather a reference to the same list object. Therefore, any changes made to

"duplicate" will also affect "collection", and vice versa. Then, the code appends the number 3 to "duplicate".

The list now contains [2, 1, 3], and so does "collection". Finally, the code tries to print the sum of the lengths of "collection" and "duplicate". However, this causes an exception, because the len function expects a single argument, not two. The code does not handle the exception, and therefore outputs nothing.

The expected output of the code is nothing, because the code raises an exception and terminates. Therefore, the correct answer is D. The code raises an exception and outputs nothing.

질문 # 22

What is the expected output of the following code?

□

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

정답: A

설명:

Explanation

The code snippet that you have sent is defining and calling a function in Python. The code is as follows:

```
def runner(brand, model, year): return (brand, model, year)
```

```
print(runner("Fermi"))
```

The code starts with defining a function called "runner" with three parameters: "brand", "model", and "year".

The function returns a tuple with the values of the parameters. A tuple is a data type in Python that can store multiple values in an ordered and immutable way. A tuple is created by using parentheses and separating the values with commas. For example, (1, 2, 3) is a tuple with three values.

Then, the code calls the function "runner" with the value "Fermi" for the "brand" parameter and prints the result. However, the function expects three arguments, but only one is given. This will cause a TypeError exception, which is an error that occurs when a function or operation receives an argument that has the wrong type or number. The code does not handle the exception, and therefore it will terminate with an error message.

However, if the code had handled the exception, or if the function had used default values for the missing parameters, the expected output of the code would be ('Fermi', '2021', 'False'). This is because the function returns a tuple with the values of the parameters, and the print function displays the tuple to the screen.

Therefore, the correct answer is D. ('Fermi', '2021', 'False').

• • • • •

PCEP-30-02인증덤프문제: <https://www.itexamdump.com/PCEP-30-02.html>

- 2026 Iteamdump 최신 PCEP-30-02 PDF 버전 시험 문제집과 PCEP-30-02 시험 문제 및 답변 무료 공유:
https://drive.google.com/open?id=1OAdX--Gs8eus-W-mXhxMM8YDWLtk_O3