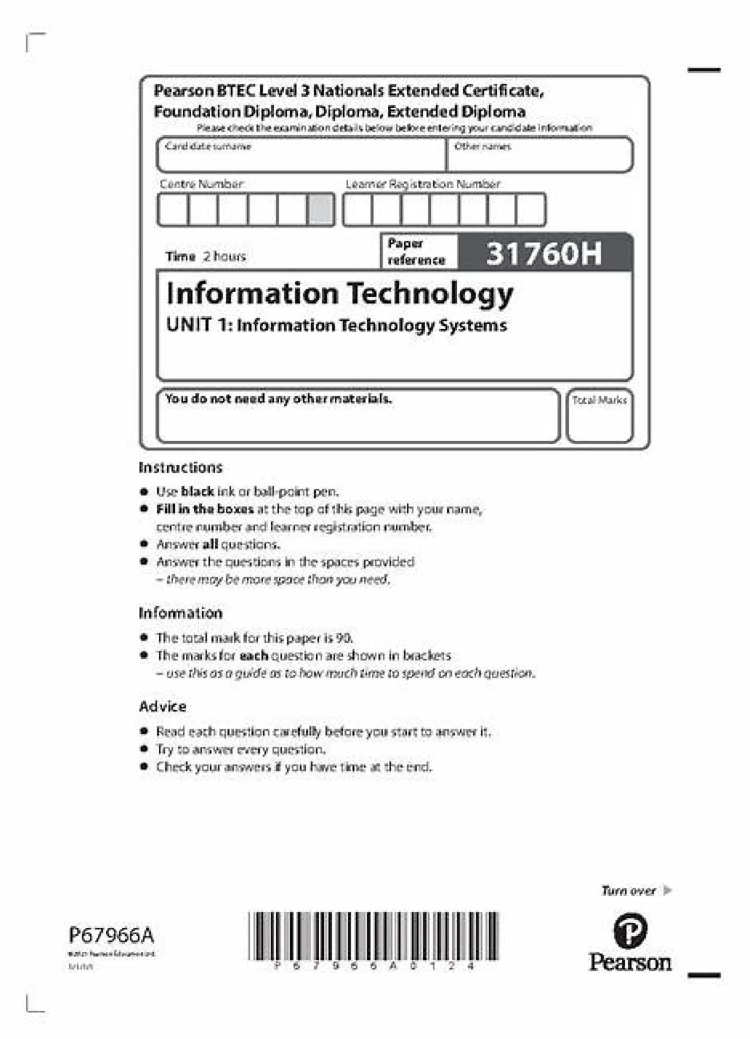
Latest Python Institute PCEP-30-02 Test Notes | PCEP-30-02 Valid Test Blueprint



Pearson BTEC Level 3 Nationals Extended Certificate, Foundation Diploma, Diploma, Extended Diploma

Please check the examination details below before entering your candidate information

Candidate surname	Other names

Centre Number □□□□□□ Learner Registration Number □□□□□□□

Time 2 hours

Paper reference **31760H**

Information Technology

UNIT 1: Information Technology Systems

You do not need any other materials.

Total Marks

Instructions

- Use **black** ink or ball-point pen.
- **Fill in the boxes** at the top of this page with your name, centre number and learner registration number.
- Answer **all** questions.
- Answer the questions in the spaces provided *– there may be more space than you need.*

Information

- The total mark for this paper is 90.
- The marks for **each** question are shown in brackets *– use this as a guide as to how much time to spend on each question.*

Advice

- Read each question carefully before you start to answer it.
- Try to answer every question.
- Check your answers if you have time at the end.

Turn over ▶

P67966A

[illegible]

P 6 7 9 6 6 A 0 1 2 4

Pearson

DOWNLOAD the newest ExamcollectionPass PCEP-30-02 PDF dumps from Cloud Storage for free: https://drive.google.com/open?id=1UyttOaMm5Z3QcHHuuEfajLtaECjLSzFy

You can learn our PCEP-30-02 test prep in the laptops or your cellphone and study easily and pleasantly as we have different types, or you can print our PDF version to prepare your exam which can be printed into papers and is convenient to make notes. Studying our PCEP-30-02 exam preparation doesn't take you much time and if you stick to learning you will finally pass the exam successfully. Believe us because the PCEP-30-02 Test Prep are the most useful and efficient, and the PCEP-30-02 exam preparation will make you master the important information and the focus of the exam. We are sincerely hoping to help you pass the exam.

If you want to pass the PCEP-30-02 exam, you should buy our PCEP-30-02 exam questions to prapare for it. Our sincerity stems from the good quality of our PCEP-30-02 learning guide is that not only we will give you the most latest content. Also we will give you one year's free update of the PCEP-30-02 Study Materials you purchase and 24/7 online service. Now just make up your mind and get your PCEP-30-02 exam braindumps!

>> Latest Python Institute PCEP-30-02 Test Notes <<

PCEP-30-02 Valid Test Blueprint & PCEP-30-02 Valid Exam Sims

Our PCEP-30-02 Exam Questions can help you pass the exam to prove your strength and increase social competitiveness.

Although it is not an easy thing for somebody to pass the PCEP-30-02 exam, but our PCEP-30-02 exam torrent can help aggressive people to achieve their goals. This is the reason why we need to recognize the importance of getting the test Python Institute certification. More qualified certification for our future employment has the effect to be reckoned with, only to have enough qualification certifications to prove their ability, can we win over rivals in the social competition.

Python Institute PCEP-30-02 Exam Syllabus Topics:

Topic	Details
Topic 1	• Loops: while, for, range(), loops control, and nesting of loops.
Topic 2	• Functions and Exceptions: This part of the exam covers the definition of function and invocation
Topic 3	• Data Collections: In this section, the focus is on list construction, indexing, slicing, methods, and comprehensions; it covers Tuples, Dictionaries, and Strings.

Python Institute PCEP - Certified Entry-Level Python Programmer Sample Questions (Q43-Q48):

NEW QUESTION # 43
What is true about exceptions and debugging? (Select two answers.)

- A. If some Python code is executed without errors, this proves that there are no errors in it.
- B. The default (anonymous) except branch cannot be the last branch in the try-except block.
- 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.

Answer: C,D

Explanation:
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.

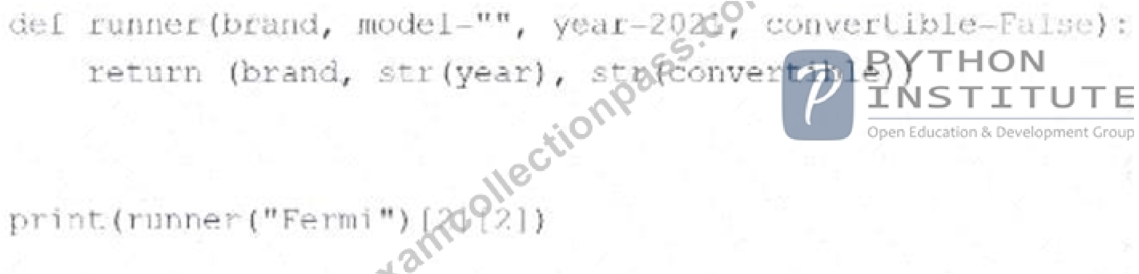
NEW QUESTION # 44
Which of the following expressions evaluate to a non-zero result? (Select two answers.)

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

Answer: B,D

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 # 45
What is the expected output of the following code?



```
def runner(brand, model="", year=2021, convertible=False):
    return (brand, str(year), str(convertible))


print(runner("Fermi")[2][2])
```

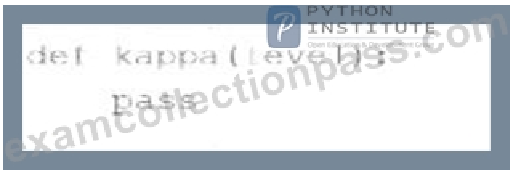
- A. 0
- B. ('Fermi ', '2021', 'False')
- C. The code raises an unhandled exception.
- D. False

Answer: B

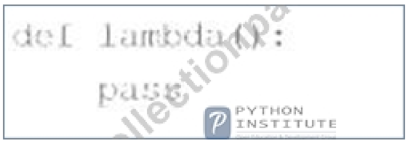
Explanation:
The code snippet that you have sent is defining and calling a function in Python. The code is as follows:
def runner(brand, model, year): return (brand, model, year)
print(runner("Fermi"))
The code starts with defining a function called "runner" with three parameters: "brand", "model", and "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').
Reference: Python Functions - W3SchoolsPython Tuples - W3SchoolsPython Exceptions: An Introduction - Real Python

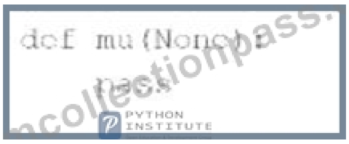
NEW QUESTION # 46
Which of the following functions can be invoked with two arguments?

- A. 

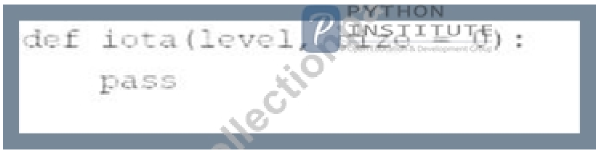
```
def kappa(level):
    pass
```

- B. 

```
def lambda():
    pass
```

- C. 

```
def mu(None):
    pass
```

- D. 

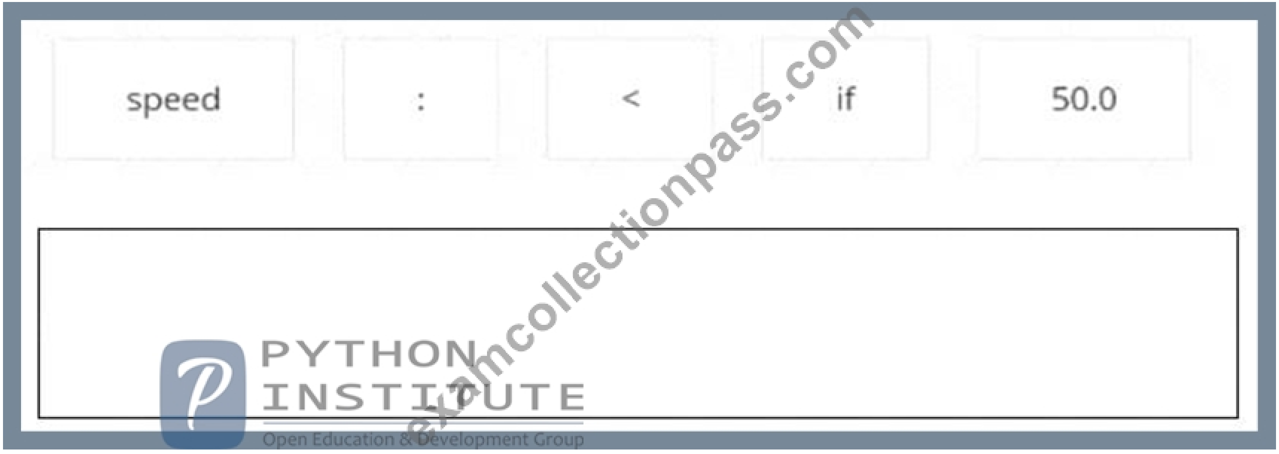
```
def iota(level, size = 0):
    pass
```

Answer: D

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.

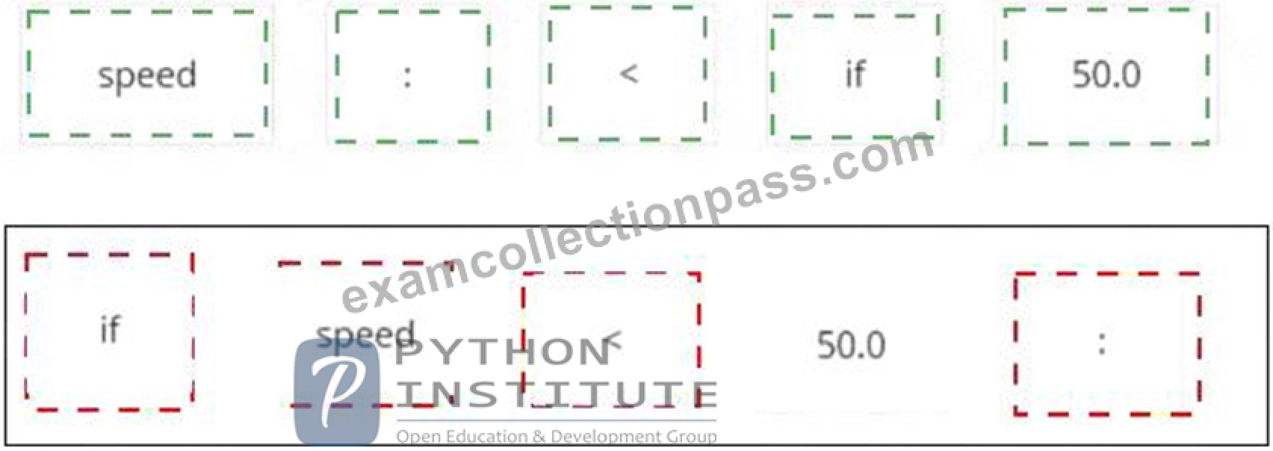
NEW QUESTION # 47

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.

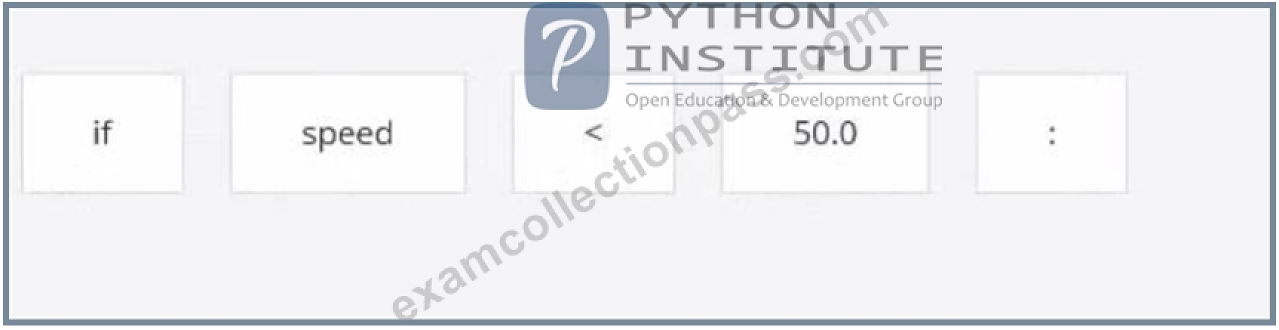


Answer:

Explanation:



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 # 48

......

The PCEP-30-02 practice questions that are best for you will definitely make you feel more effective in less time. The cost of PCEP-30-02 studying materials is really very high. Selecting our study materials is definitely your right decision. Of course, you can also make a decision after using the trial version. With our PCEP-30-02 Real Exam, we look forward to your joining. And our PCEP-30-02 exam braindumps will never let you down.

PCEP-30-02 Valid Test Blueprint: https://www.examcollectionpass.com/Python-Institute/PCEP-30-02-practice-exam-dumps.html

- Practical Latest PCEP-30-02 Test Notes | Easy To Study and Pass Exam at first attempt - Efficient Python Institute PCEP - Certified Entry-Level Python Programmer □ Search for （ PCEP-30-02 ） and download exam materials for free through □ www.prepawaypdf.com □ □Reliable PCEP-30-02 Real Test
- PCEP-30-02 Pass4sure Exam Prep □ PCEP-30-02 Latest Braindumps Ebook □ Reliable PCEP-30-02 Braindumps Ppt □ Download ➠ PCEP-30-02 □ for free by simply entering ▶ www.pdfvce.com ◀ website □PCEP-30-02 Latest Test Bootcamp
- 2026 Unparalleled Latest PCEP-30-02 Test Notes Help You Pass PCEP-30-02 Easily □ The page for free download of { PCEP-30-02 } on ✔ www.examcollectionpass.com □✔□ will open immediately □Detail PCEP-30-02 Explanation
- Detail PCEP-30-02 Explanation □ PCEP-30-02 Latest Exam Testking □ PCEP-30-02 Download Demo □ Open [www.pdfvce.com] enter 「 PCEP-30-02 」 and obtain a free download □Reliable PCEP-30-02 Real Test
- PCEP-30-02 Test Guide: Python Institute PCEP - PCEP-30-02 Exam Torrent - PCEP-30-02 Training Materials □ Open website ➥ www.testkingpass.com □ and search for ➽ PCEP-30-02 □ for free download ✍Detail PCEP-30-02 Explanation
- PCEP-30-02 Exam Topics □ Practice PCEP-30-02 Questions □ Detail PCEP-30-02 Explanation □ Open [www.pdfvce.com] and search for □ PCEP-30-02 □ to download exam materials for free □Detail PCEP-30-02 Explanation
- PCEP-30-02 Test Guide: Python Institute PCEP - PCEP-30-02 Exam Torrent - PCEP-30-02 Training Materials □ Search for □ PCEP-30-02 □ and obtain a free download on ➥ www.practicevce.com □ □PCEP-30-02 Download Demo
- PCEP-30-02 Pass4sure Exam Prep □ PCEP-30-02 Valid Exam Pass4sure □ PCEP-30-02 Latest Braindumps Ebook □ Copy URL ➠ www.pdfvce.com □ open and search for ➥ PCEP-30-02 □ to download for free □Practice PCEP-30-02 Questions
- Practice PCEP-30-02 Questions □ PCEP-30-02 Latest Exam Testking □ PCEP-30-02 Passleader Review □ Search for ▷ PCEP-30-02 ◁ and download it for free on ▷ www.troytecdumps.com ◁ website □PCEP-30-02 Reliable Exam Registration
- Latest PCEP-30-02 Test Notes - Leading Provider in Qualification Exams - PCEP-30-02 Valid Test Blueprint □ Simply search for 《 PCEP-30-02 》 for free download on ➠ www.pdfvce.com □ □Reliable PCEP-30-02 Braindumps Ppt
- Latest PCEP-30-02 Learning Materials □ Dump PCEP-30-02 File □ Exam PCEP-30-02 Topics □ Open website □ www.practicevce.com □ and search for ✔ PCEP-30-02 □✔□ for free download □PCEP-30-02 Reliable Exam Registration
- myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, kumu.io, www.stes.tyc.edu.tw, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, www.stes.tyc.edu.tw, myportal.utt.edu.tt, vibelearny.com, myportal.utt.edu.tt, Disposable vapes

2026 Latest ExamcollectionPass PCEP-30-02 PDF Dumps and PCEP-30-02 Exam Engine Free Share: https://drive.google.com/open?id=1UyttOaMm5Z3QcHHuuEfajLtaECjLSzFy