Python-Institute

Exam Questions PCEP-30-02

PCEP - Certified Entry-Level Python Programmer

About Exambible

Your Partner of IT Exam

Found in 1998

Exambible is a company specialized on providing high quality IT exam practice study materials, especially Cisco CCNA, CCDA, CCNP, CCIE, Checkpoint CCSE, CompTIA A+, Network+ certification practice exams and so on. We guarantee that the candidates will not only pass any IT exam at the first attempt but also get profound understanding about the certificates they have got. There are so many alike companies in this industry, however, Exambible has its unique advantages that other companies could not achieve.

Our Advances

* 99.9% Uptime

All examinations will be up to date.

* 24/7 Quality Support

We will provide service round the clock.

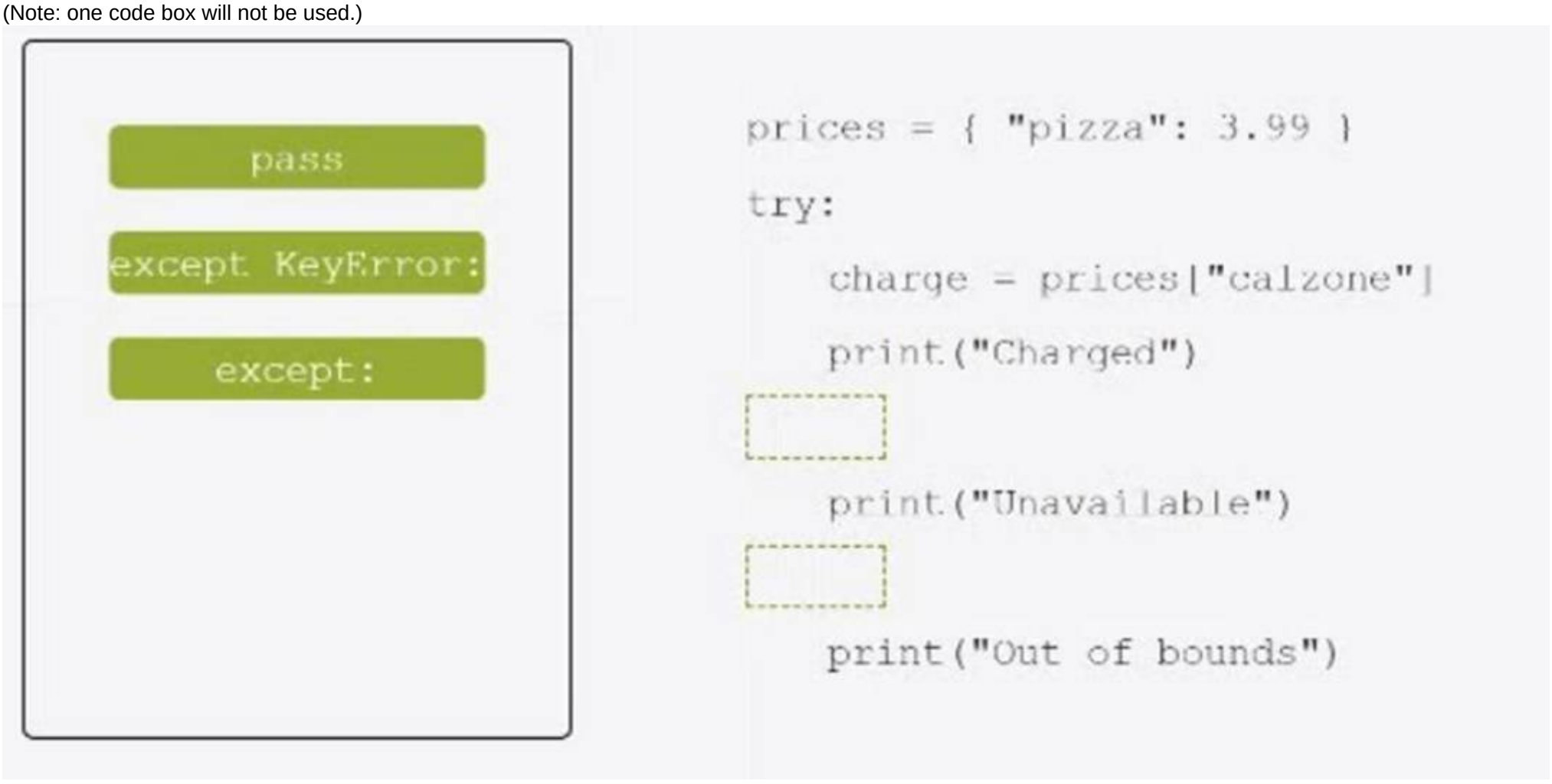
* 100% Pass Rate

Our guarantee that you will pass the exam.

* Unique Gurantee

If you do not pass the exam at the first time, we will not only arrange FULL REFUND for you, but also provide you another exam of your claim, ABSOLUTELY FREE!

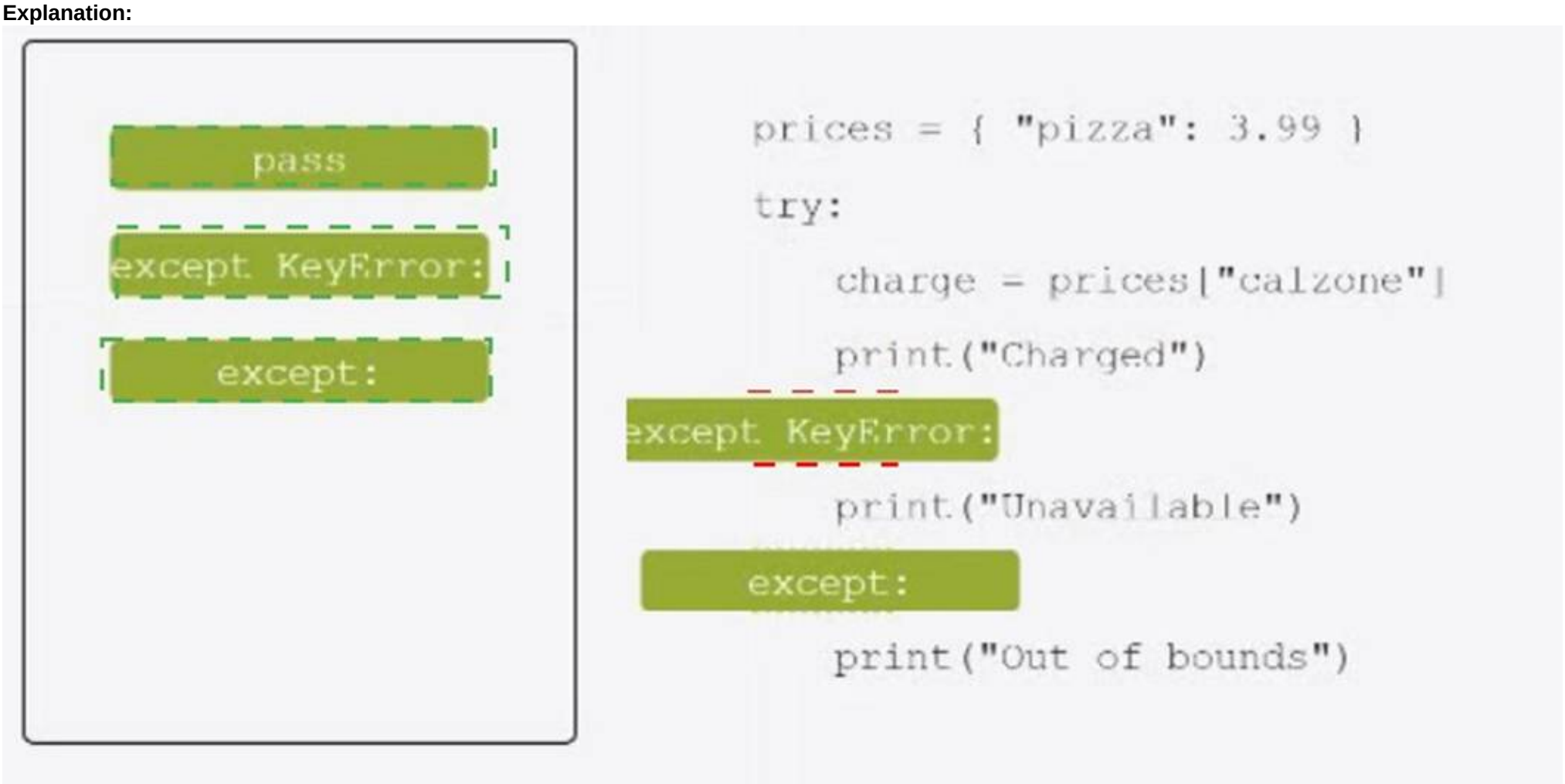
NEW QUESTION 1
DRAG DROP
Drag and drop the code boxes in order to build a program which prints Unavailable to the screen.
(Note: one code box will not be used.)



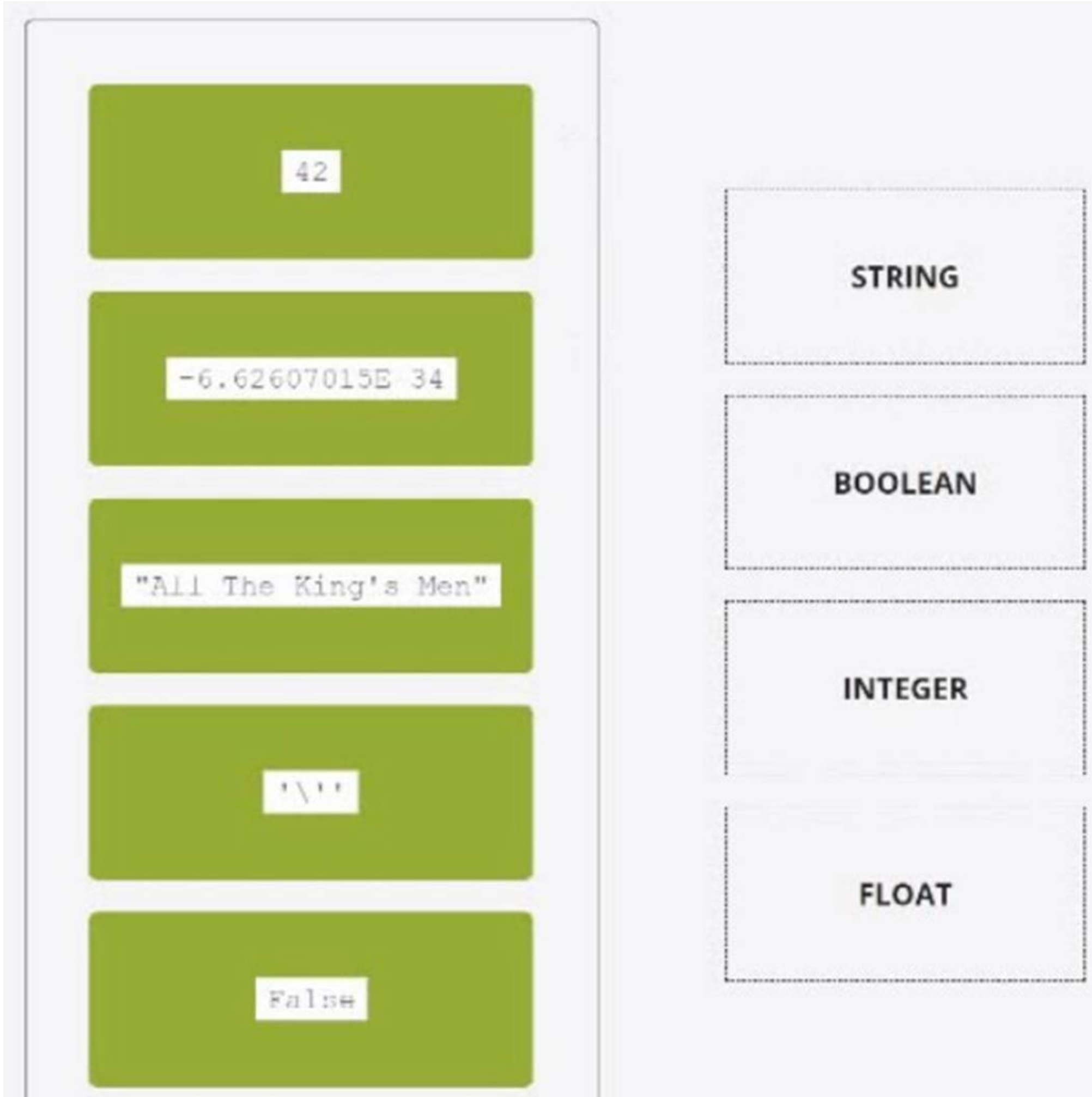
A. Mastered
B. Not Mastered

Answer: A

Explanation:



NEW QUESTION 2
DRAG DROP
Drag and drop the literals to match their data type names.



A. Mastered
B. Not Mastered

Answer: A

Explanation:

One possible way to drag and drop the literals to match their data type names is:
? STRING: ??All The King??s Men??
? BOOLEAN: False
? INTEGER: 42
? FLOAT: -6.62607015E-34
A literal is a value that is written exactly as it is meant to be interpreted by the Python interpreter. A data type is a category of values that share some common characteristics or operations. Python has four basic data types: string, boolean, integer, and float.
A string is a sequence of characters enclosed by either single or double quotes. A string can represent text, symbols, or any other information that can be displayed as text. For example, ??All The King??s Men?? is a string literal that represents the title of a novel.
A boolean is a logical value that can be either True or False. A boolean can represent the result of a comparison, a condition, or a logical operation. For example, False is a boolean literal that represents the opposite of True.
An integer is a whole number that can be positive, negative, or zero. An integer can represent a count, an index, or any other quantity that does not require fractions or decimals. For example, 42 is an integer literal that represents the answer to life, the universe, and everything.
A float is a number that can have a fractional part after the decimal point. A float can represent a measurement, a ratio, or any other quantity that requires precision or

approximation. For example, -6.62607015E-34 is a float literal that represents the Planck constant in scientific notation.
You can find more information about the literals and data types in Python in the following references:
? [Python Data Types]
? [Python Literals]
? [Python Basic Syntax]

NEW QUESTION 3
What is the expected output of the following code?

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

A. 5
B. 4
C. 6
D. The code raises an exception and outputs nothing.

Answer: D

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

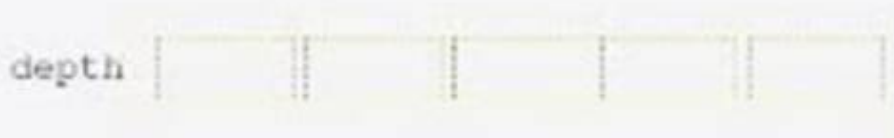
NEW QUESTION 4
A set of rules which defines the ways in which words can be coupled in sentences is called:

A. lexis
B. syntax
C. semantics
D. dictionary

Answer: B

Explanation:
Syntax is the branch of linguistics that studies the structure and rules of sentences in natural languages. Lexis is the vocabulary of a language. Semantics is the study of meaning in language. A dictionary is a collection of words and their definitions, synonyms, pronunciations, etc.
Reference: [Python Institute - Entry-Level Python Programmer Certification]

NEW QUESTION 5
DRAG DROP
Insert the code boxes in the correct positions in order to build a line of code which asks the user for an integer value and assigns it to the depth variable.
(Note: some code boxes will not be used.)



A. Mastered
B. Not Mastered

Answer: A

Explanation:

One possible way to insert the code boxes in the correct positions in order to build a line of code which asks the user for an integer value and assigns it to the depth variable is:
depth = int(input("Enter the immersion depth: "))
This line of code uses the input function to prompt the user for a string value, and then uses the int function to convert that string value into an integer number. The result is then assigned to the variable depth.
You can find more information about the input and int functions in Python in the following references:
? [Python input() Function]
? [Python int() Function]

NEW QUESTION 6
What happens when the user runs the following code?

```
speed = 0
while speed < 30:
    speed *= 2
    if speed > 10:
        continue
    print("*", end="")
else:
    print("*")
```

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

Answer: D

Explanation:
The code snippet that you have sent is a while loop with an if statement and a print statement inside it. The code is as follows:
while True: if counter < 0: print(????) else: print(??**??)
The code starts with entering a while loop that repeats indefinitely, because the condition ??True?? is always true. Inside the loop, the code checks if the value of ??counter?? is less than 1. If yes, it prints a single asterisk () to the screen. If no, it prints three asterisks (**) to the screen. However, the code does not change the value of ??counter?? inside the loop, so the same condition is checked over and over again. The loop never ends, and the code enters an infinite loop.
The program outputs either one asterisk () or three asterisks (**) to the screen repeatedly, depending on the initial value of ??counter??. Therefore, the correct answer is D. The program enters an infinite loop.
Reference: [Python Institute - Entry-Level Python Programmer Certification]

NEW QUESTION 7
What is the expected output of the following code?

```
def traverse(stop):
    if stop == 0:
        return 0
    else:
        return stop * traverse(stop - 1)


print(traverse(2))
```

A. 2
B. 3
C. 1

Answer: D

Explanation:
The code snippet that you have sent is using the count method to count the number of occurrences of a value in a list. The code is as follows:
my_list = [1, 2, 3, 4, 5] print(my_list.count(1))
The code starts with creating a list called ??my_list?? that contains the numbers 1, 2, 3, 4, and 5. Then, it uses the print function to display the result of calling the count method on the list with the argument 1. The count method is used to return the number of times a value appears in a list. For example, my_list.count(1) returns 1, because 1 appears once in the list.
The expected output of the code is 1, because the code prints the number of occurrences of 1 in the list. Therefore, the correct answer is D. 1.
Reference: Python List count() Method - W3Schools

NEW QUESTION 8
Python Is an example of which programming language category?

A. interpreted
B. assembly
C. compiled
D. machine

Answer: A

Explanation:
Python is an interpreted programming language, which means that the source code is translated into executable code by an interpreter at runtime, rather than by a compiler beforehand. Interpreted languages are more flexible and portable than compiled languages, but they are also slower and less efficient. Assembly and machine languages are low-level languages that are directly executed by the hardware, while compiled languages are high-level languages that are translated into machine code by a compiler before execution.
Reference: [Python Institute - Entry-Level Python Programmer Certification]

NEW QUESTION 9
What is the expected result of the following code?

```
def velocity(x-10):
    return speed | x


speed = 10
new_speed = velocity()
new_speed = velocity(new_speed)
print(new_speed)
```

A. The code is erroneous and cannot be run.
B. 20
C. 10
D. 30

Answer: A

Explanation:
The code snippet that you have sent is trying to use the global keyword to access and modify a global variable inside a function. The code is as follows:
speed = 10 def velocity(): global speed speed = speed + 10 return speed print(velocity())
The code starts with creating a global variable called ??speed?? and assigning it the value 10. A global variable is a variable that is defined outside any function and can be accessed by
any part of the code. Then, the code defines a function called ??velocity?? that takes no parameters and returns the value of ??speed?? after adding 10 to it. Inside the function, the code uses the global keyword to declare that it wants to use the global variable ??speed??, not a local one. A local variable is a variable that is defined inside a function and can only be accessed by that function. The global keyword allows the function to modify the global variable, not just read it. Then, the code adds 10 to the value of ??speed?? and returns it. Finally, the code calls the function ??velocity?? and prints the result.
However, the code has a problem. The problem is that the code uses the global keyword inside the function, but not outside. The global keyword is only needed

when you want to modify a global variable inside a function, not when you want to create or access it outside a function. If you use the global keyword outside a function, you will get a SyntaxError exception, which is an error that occurs when the code does not follow the rules of the Python language. 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 uses the global keyword incorrectly. Therefore, the correct answer is A. The code is erroneous and cannot be run.
Reference: Python Global Keyword - W3SchoolsPython Exceptions: An Introduction – Real
Python
The code is erroneous because it is trying to call the ??velocity?? function without passing any parameter, which will raise a TypeError exception. The ??velocity?? function requires one parameter ??x??, which is used to calculate the return value of ??speed?? multiplied by ??x??. If no parameter is passed, the function will not know what value to use for ??x??.
The code is also erroneous because it is trying to use the ??new_speed?? variable before it is defined. The ??new_speed?? variable is assigned the value of 20 after the first function call, but it is used as a parameter for the second function call, which will raise a NameError
exception. The variable should be defined before it is used in any expression or function call.
Therefore, the code will not run and will not produce any output. The correct way to write the code would be:
Define the speed variable speed = 10
Define the velocity function def velocity(x):
return speed * x
Define the new_speed variable new_speed = 20
Call the velocity function with new_speed as a parameter print(velocity(new_speed))
Copy
This code will print 200, which is the result of 10 multiplied by 20. References:
[Python Programmer Certification (PCPP) – Level 1] [Python Programmer Certification (PCPP) – Level 2] [Python Programmer Certification (PCPP) – Level 3]
[Python: Built-in Exceptions]
[Python: Defining Functions]
[Python: More on Variables and Printing]

NEW QUESTION 10
What is the expected output of the following code?

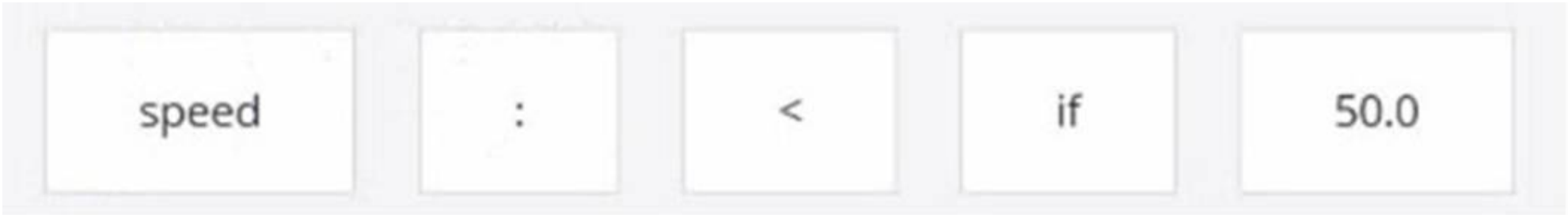
```
menu = {"pizza": 2.39, "pasta": 1.99, "folpetti": 3.99}

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 10
DRAG DROP
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.



A. Mastered
B. Not Mastered

Answer: A

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 13
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 16
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. 1
B. The code raises an unhandled exception.
C. False
D. ('Fermi ', '2021', 'False')

Answer: D

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 18
Which of the following functions can be invoked with two arguments?
A)

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

B)

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

C)

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

D)

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

A. Option A
B. Option B
C. Option C
D. Option 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 23
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 24
.....

Relate Links

100% Pass Your PCEP-30-02 Exam with Exambible Prep Materials

https://www.exambible.com/PCEP-30-02-exam/

Contact us

We are proud of our high-quality customer service, which serves you around the clock 24/7.

Viste - https://www.exambible.com/