Vendor: Python Institute

Exam Code:PCEP-30-02

Exam Name:PCEP - Certified Entry-Level Python Programmer

Version: Demo

QUESTION 1

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

Correct 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]

QUESTION 2

What is the expected output of the following code?

marks = [80, 70, 90, 90, 80, 100]

average = sum(marks) // len(marks)

grade = \\'\\'

```
marks = [80, 70, 90, 90, 80, 100]
average = sum(marks) // len(marks)
grade = ''

if 90 <= average <= 100:
    grade = 'A'
elif 80 <= average < 90:
    grade = 'B'
elif 70 <= average < 80:
    grade = 'C'
elif 65 <= average < 70:
    grade = 'D'
else:
    grade = 'F'

print(grade)       # B
print(sum(marks))  # 510
print(average)     # 85
```

A. A

B. C

C. The code is erroneous.

D. D

E. B

F. F

Correct Answer: E

QUESTION 3

Only one of the following statements is true - which one?

A. addition precedes multiplication

B. multiplication precedes addition

C. neither statement can be evaluated

Correct Answer: B

QUESTION 4

You are an intern for ABC electric cars company. You must create a function that calculates the average velocity of their vehicles on a 1320 foot (1/4 mile) track. Consider the following code.

```
# distance = float(input('Enter the distance travelled in feet'))
distance = float('437723.42')  # Just for convenience
distance_miles = distance/5280

# time = float(input('Enter the time elapsed in seconds'))
time = float('1723.9')  # Just for convenience
time_hours = time/3600  # convert to hours

velocity = distance_miles/time_hours
print('The average Velocity : ', velocity, 'miles/hour')
# The average Velocity : 173.1236071486956 miles/hour

print((float('437723.42')/5280) / (float('1723.9')/3600))
# 173.1236071486956
# print((int('437723.42')/5280) / (int('1723.9')/3600))
# ValueError
print((int(float('437723.42'))/5280) / (int(float('1723.9'))/3600))
# 173.21387115496228
```

The output must be as precise as possible. What would you insert instead of ??? and ??? ?

A. 1 | int 2 | int

B. 1 | oat 2 | oat

C. 1 | int 2 | oat

D. 1 | oat 2 | int

Correct Answer: B

QUESTION 5

UTF 8 is ...

A. the 9th version of the UTF Standard.

B. a synonym for "byte".

C. a Python version name.

D. an encoding form of the Unicode Standard.

Correct Answer: D

QUESTION 6

What is the expected behavior of the following program?

```
1  prin("Goodbye!")
```

A. The program will generate an error message on the screen

B. The program will output "Goodbye!"

C. The program will output Goodbye!

D. The program will output ("Goodbye!")

Correct Answer: A

QUESTION 7

What is the expected output of the following code?

```
data = {'one': 'two', 'two': 'three', 'three': 'one'}
res = data['three']

for _ in range(len(data)):
    res = data[res]

print(res)
```

A. three

B. (\\'one\\', \\'two\\', \\'three\\')

C. two

D. one

Correct Answer: D

QUESTION 8

Which of the following snippets shows the correct way of handling multiple exceptions in a single except clause?

A.
```
except TypeError, ValueError, ZeroDivisionError
    # Some code.
```

B.
```
except (TypeError, ValueError, ZeroDivisionError):
    # Some code.
```

C.
```
except (TypeError, ValueError, ZeroDivisionError)
    # Some code.
```

D.
```
except: TypeError, ValueError, ZeroDivisionError
    # Some code.
```

E.
```
except: (TypeError, ValueError, ZeroDivisionError)
    # Some code.
```

F.
```
try:
    print(7 / 0)
except (TypeError, ValueError, ZeroDivisionError):
    print("That is not allowed!")
```

A. Option A

B. Option B

C. Option C

D. Option D

E. Option E

Correct Answer: C

QUESTION 9

What is the output of the following snippet?

```
1  def fun(x, y, z):
2      return x + 2 * y + 3 * z
3
4  print(fun(0, z=1, y=3))
```

A. 9

B. The snippet is erroneous.

C. 0

D. 3

Correct Answer: A

QUESTION 10

A code point is:

A. A point used to write a code.

B. A number which makes up a character.

C. A code containing a point.

Correct Answer: B

QUESTION 11

If you want to build a string that reads:

Peter\\'s sister\\'s name\\'s "Anna"

Which of the following literals would you use? (Choose two.)

A. \\'Peter\\'s sister\\'s name\\'s "Anna"\\'

B. \\'Peter\\\'s sister\\\'s name\\\'s \"Anna\"\\'

C. "Peter\\'s sister\\'s name\\'s "Anna""

D. "Peter\\'s sister\\'s name\\'s \"Anna\""

Correct Answer: BD

QUESTION 12

The following is a program to validate customer numbers.

```
# customer_number = input('Enter the employee number (dd-ddd-dddd): ')
customer_number = '12-345-6789'     # True
# customer_number = '12345-6789'    # False
# customer_number = 'A2-345-6789'   # False
# customer_number = '112-345-6789'  # False
parts = customer_number.split('-')
valid = False
if len(parts) == 3:
  if len(parts[0]) == 2 and len(parts[1]) == 3 and len(parts[2]) == 4:
    if parts[0].isdigit() and parts[1].isdigit() and
parts[2].isdigit():
      valid = True
print(valid)

# You might have to scroll a little to the right to see everything
```

The number may only contain numbers and dashes. The number must have the right format (dd-ddd-dddd). What is true about this program?

A. There will be no error but there will be an unwanted result.

B. There will be a SyntaxError.

C. The program works properly.

D. There will be an AttributeError.

Correct Answer: C