CoderSchool

M2.1 Python 101

Exam Solution

Part 1: MULTIPLE CHOICES

Question 1: Given the following code:

```
name = "Tom"
height = 180
```

Which of the following print statements will give an error ?

```
A. print("My name is " + name + " and my height is " + height)
```

Question 2: Given the following code:

```
def foo(x,y):
    x = x+1
    z = 3*x + 2*y
z = 10
foo(2, 1)
```

What is the value of **z** after running the code above?

```
B. 10
```

Question 3: Given the following code:

```
a = "Hello World Philippines!"
```

Which of the following are empty strings?

```
c = a[:-1:-1]
e = a[1:-2:-1]
```

Question 4: Given the following code:

```
list_1 = [0,1,2,3,4,5]
list_2 = list_1
list_2[0] +=1
list_2 = list_2.pop(5)
```

What is the value of list_1 after running the code above?

```
B. [1, 1, 2, 3, 4]
```

Question 5: Given the following code:

```
w = "good morning Mariana"
v = ('a','e','i','o','u')
t = "aeiou"
l = list(v)
s = set(t)
```

Which list comprehension returns a list of all vowels in the string w?

`E. All of the above`

Question 6: Given the following code:

```
s = ' _-_CoderSchool Data Science 2022 _ _'
```

Which of the following expressions correctly returns the string "CoderSchool Data Science 2022"?

`A. s.strip('-_ ')`

`C. s.replace('_','').replace('-','').strip()`

Question 7: Which of the following function definitions is VALID (i.e. run without error)?

Note: this question is not asking if the function works as intended, only if the function definition runs without error.

C.

```
def mult_3_plus_1(x):
    y = 3*x
    z = y+1
    return z
```

D.

```
def sum_of_first_n_numbers():
    n = int(input())
    my_sum, i = 0, 1
    while i <= n:
        my_sum += i
    print(f'Sum of first {n} numbers:', my_sum)
```

Question 8: Remember we have learned about dictionaries in python, we know every entry in a dictionary will have a key and a value.

Which statement is FALSE about dictionary keys and values?

`C. Keys must be string.`

Part 2: FUNCTIONS

Question 9:

```
def count_min(my_list):
    return my_list.count(min(my_list))
```

Question 10:

```
def calculate_range(my_tup):
    return max(my_tup) - min(my_tup)
```

Question 11:

```
def extract_email(email, get_username):
    return email.split('@')[0] if get_username else email.split('@')[1]
```

Question 12:

```
def item_calculator(item, get_weight):
    weight = item['unit_weight'] * item['number_of_units']
    cost   = item['unit_price'] * item['number_of_units']
    return weight if get_weight else cost
```

Question 13:

```
def heaviest_item(receipt):
    def item_calculator(item, get_weight):
        weight = item['unit_weight'] * item['number_of_units']
        cost   = item['unit_price'] * item['number_of_units']
        return weight if get_weight else cost
    weight_receipt = {
        item:item_calculator(item_info, True)
        for item, item_info in receipt.items()
    }
    return max(weight_receipt, key=weight_receipt.get)
```

Question 14:

```
def priciest_item(receipt):
    def item_calculator(item, get_weight):
        weight = item['unit_weight'] * item['number_of_units']
        cost   = item['unit_price'] * item['number_of_units']
        return weight if get_weight else cost
    price_receipt = {item:item_calculator(item_info, False) for item,
item_info in receipt.items()}
    return max(price_receipt, key=price_receipt.get)
```