Exam 2

ⓘ This is a preview of the published version of the quiz

Started: Mar 1 at 1:13pm

Quiz Instructions

Question 1

1 pts

Using asserts is **most useful** for dealing with which category of errors?

○ semantic

○ runtime

○ syntax

Question 2

1 pts

What will be printed after the following is executed?

```
val = "umbrella"
count = dict()
for c in val:
    if not c in count:
        count[c] = 0
    else:
        count[c] += 2
print(count["l"])
```

○ 0

○ 4

○ 1

○ 2

Question 3

1 pts

What will be in the file after this code runs? The file exists when the code starts.

```
f = open("file.txt", "w")
f.write("Hello")
f.close()
f = open("file.txt", "w")
f.write("CS220 students")
f.close()
```

○ Hello CS220 students

○ CS220 students

○ HelloCS220 students

○ Hello

Question 4

1 pts

What will be printed after execution?

```
d = ['a']
a = ['b']
b = ['d']
c = a
c.append(d)
print(a,c)
```

○ ['a','c','d']

○ ['b'] ['b', ['a']]

○ ['b', ['a']] ['b', ['a']]

○ ['a','c']

○ ['a','b','c']

Question 5

1 pts

True or False question:

If a function defined in Python doesn't have an explicit "return" statement, it automatically returns the last variable present in the body of the function.

○ True

○ False

Question 6

1 pts

What will be printed by the following segment of code?

```
name = "Data Programming Works!"
words = name.split(" ")
print(words.pop(-1))
```

○ Works

○ Data Programming

○ Works!

○ !

Question 7

1 pts

What type of objects are immutable in Python?

□ tuple

□ Custom types created using namedtuple

□ Lists

□ Custom types created using recordclass

□ Strings

□ Dictionaries

Question 8

1 pts

Which among the following is an immutable version of lists?

○ tuples

○ namedtuples

○ sets

○ recordclass

Question 9

1 pts

Which of the following cannot be used as a key for a Python dictionary?

○ strings

○ negative integers

○ tuples

○ namedtuples

○ lists

Question 10

1 pts

What is the output of the following code?

```
sorted(["f", "E", "h", "G"], reverse = True)
```

○ ['h', 'G', 'f', 'E']

○ ['E', 'G', 'f', 'h']

○ ['h', 'f', 'G', 'E']

○ ['E', 'f', 'G', 'h']

Question 11

1 pts

Consider the following code snippet and then answer the question below it.

```
header = ['room_id',  'name', 'neighborhood_group', 'latitude', 'number_of_revie
ws']
rooms = [
['11943', 'Country space in the city', 'Brooklyn', 40.63702,0 ],
['206071',  'Yankee Stadium Oasis 2 stops to Manhattan!',  'Bronx',  40.82802, 25
3],
['24285', 'Beautiful Duplex Apartment',  'Brooklyn', 40.66941, 30],
['151199', 'Astoria-Private Home NYC-', 'Queens', 40.75725, 414],
['98330', 'LOVELY APARTMENT IN THE HEART OF NY', 'Manhattan', 40.73877, 1],
['60611', 'SpaHa Studio Monthly Rental', 'Manhattan', 40.79163, 183]
]
```

What does the following evaluate to?

```
rooms[3][:2]
```

○ ['151199', 'Astoria-Private Home NYC-', 'Queens']

○ [40.66941, 30]

○ []

○ ['24285', 'Beautiful Duplex Apartment']

○ ['151199', 'Astoria-Private Home NYC-']

Question 12

1 pts

Consider the following code snippet and then answer the question below it.

```
id_to_name = {
11943: 'Country space in the city',
206071:  'Yankee Stadium Oasis 2 stops to Manhattan!',
24285: 'Beautiful Duplex Apartment',
151199: 'Astoria-Private Home NYC-',
98330: 'LOVELY APARTMENT IN THE HEART OF NY',
60611: 'SpaHa Studio Monthly Rental'
}
```

What is the output for:

```
print (type(id_to_name), type(id_to_name.get(151199)), type(id_to_name.get("9833
0")), type(id_to_name.get("60611", 10.02)))
```

○ <class 'dict'>, <class 'int'>, <class 'NoneType'>, <class 'float'>

○ <class 'dict'>, <class 'str'>, <class 'NoneType'>, <class 'str'>

○ <class 'dict'>, <class 'str'>, <class 'NoneType'>, <class 'float'>

○ <class 'dict'>, <class 'int'>, <class 'str'>, <class 'NoneType'>

Question 13

1 pts

Consider the following code snippet and then answer the question below it.

```
id_to_name = {
'11943': 'Country space in the city',
'206071':  'Yankee Stadium Oasis 2 stops to Manhattan!',
'24285': 'Beautiful Duplex Apartment',
'151199': 'Astoria-Private Home NYC-',
'98330': 'LOVELY APARTMENT IN THE HEART OF NY',
'60611': 'SpaHa Studio Monthly Rental'
}
```

What is the output for len(id_to_name)?

○ 6

○ 10

○ 5

○ 0

○ len doesn't work for dict.

○ 12

Question 14

1 pts

Consider the following code snippet and answer the question below it.

```
header = ['room_id',  'name', 'neighborhood_group', 'latitude', 'number_of_revie
ws']
rooms = [
['11943', 'Country space in the city', 'Brooklyn', 40.63702,0 ],
['206071',  'Yankee Stadium Oasis 2 stops to Manhattan!',  'Bronx',  40.82802, 25
3],
['24285', 'Beautiful Duplex Apartment',  'Brooklyn', 40.66941, 30],
['151199', 'Astoria-Private Home NYC-', 'Queens', 40.75725, 414],
['98330', 'LOVELY APARTMENT IN THE HEART OF NY', 'Manhattan', 40.73877, 1],
['60611', 'SpaHa Studio Monthly Rental', 'Manhattan', 40.79163, 183]
]

def get_ids(rooms):
    all_ids = []
    for i in range(len(rooms)):
        all_ids.append(????)
    return all_ids
```

The function get_ids(rooms) must return a list of all room_ids from the dataset rooms. What should ???? be replaced with?

○ row[header.index("room_id")]

○ rooms[i][header.index("room_id")]

○ rooms[i]["room_id"]

○ row[i]

Question 15

1 pts

Consider the following code snippet and then answer the question below it.

```
id_to_name = {
'11943': 'Country space in the city',
'206071':  'Yankee Stadium Oasis 2 stops to Manhattan!',
'24285': 'Beautiful Duplex Apartment',
'151199': 'Astoria-Private Home NYC-',
'98330': 'LOVELY APARTMENT IN THE HEART OF NY',
'60611': 'SpaHa Studio Monthly Rental'
}
```

The right syntax to iterate over the keys in the dictionary id_to_name is:

○ for key,value in id_to_name:

○ for key in id_to_name.values():

○ for key in id_to_name:

○ for i in range(len(id_to_name)):

Question 16

1 pts

Consider the following code snippet. Assume that this code snippet is executed and then answer the question below it.

```
movies = [
    {"title": "A", "year": 19, "style": "long", "genres": ["g1", "g2"]},
    {"title": "B", "year": 19, "style": "short", "genres": ["g2", "g3"]},
    {"title": "C", "year": 20, "style": "short", "genres": ["g1", "g3"]},
    {"title": "D", "year": 20, "style": "long", "genres": ["g1", "g2", "g3"]},
    {"title": "E", "year": 20, "style": "long", "genres": ["g2"]}
]
```

What would the following code print?

```
a = movies.pop(-1)
b = a.pop("long")
print(b)
```

○ "style"

○ KeyError: "long"

○ {"title": "E", "year": 20, "style": "long", "genres": ["g2"]}

○ None

○ []

Question 17

1 pts

Consider the following code snippet. Assume that this code snippet is executed and then answer the question below it.

```
movies = [
    {"title": "A", "year": 19, "style": "long", "genres": ["g1", "g2"]},
    {"title": "B", "year": 19, "style": "short", "genres": ["g2", "g3"]},
    {"title": "C", "year": 20, "style": "short", "genres": ["g1", "g3"]},
    {"title": "D", "year": 20, "style": "long", "genres": ["g1", "g2", "g3"]},
    {"title": "E", "year": 20, "style": "long", "genres": ["g2"]}
]
```

What will counts contain? Be careful!

```
counts = {}
for m in movies:
    for genre in m["genres"]:
        if not genre in counts:
            counts[genre] = 1
        counts[genre] += 1
```

○ {"g1": 3, "g2": 3, "g3": 3}

○ {}

○ {"g1": 3, "g2": 3}

○ {"g1": 3, "g2": 4, "g3": 3}

○ {"g1": 4, "g2": 5, "g3": 4}

Question 18 1 pts

Consider the following code snippet. Assume that this code snippet is executed and then answer the question below it.

```
movies = [
    {"title": "A", "year": 19, "style": "long", "genres": ["g1", "g2"]},
    {"title": "B", "year": 19, "style": "short", "genres": ["g2", "g3"]},
    {"title": "C", "year": 20, "style": "short", "genres": ["g1", "g3"]},
    {"title": "D", "year": 20, "style": "long", "genres": ["g1", "g2", "g3"]},
    {"title":  "E", "year":  20, "style":  "long", "genres":  ["g2"]}
]
```

What will be printed? Be careful!

```
buckets = {}
bucket = []
for m in movies:
    key = m["year"]
    if not key in buckets:
        buckets[key] = bucket
    buckets[key].append(key)
print(len(buckets[19]))
```

○ 3

○ 1

○ 2

○ 5

○ 4

Question 19

1 pts

Consider the following code snippet. Assume that this code snippet is executed and then answer the question below it.

```
movies = [
    {"title":  "A", "year":  19, "style":  "long", "genres":  ["g1", "g2"]},
    {"title":  "B", "year":  19, "style":  "short", "genres":  ["g2", "g3"]},
    {"title":  "C", "year":  20, "style":  "short", "genres":  ["g1", "g3"]},
    {"title":  "D", "year":  20, "style":  "long", "genres":  ["g1", "g2", "g3"]},
    {"title":  "E", "year":  20, "style":  "long", "genres":  ["g2"]}
]
```

How many lines will be printed to the screen?

```
for i in range(10):
    try:
        m = movies[i]
        print(m["title"])
        year = m["DATE"]
        print(year)
    except:
        print("Exception occurred!")
        print("What happened to me?")
```

○ 10

○ 20

○ 5

○ 15

○ 25

Question 20

1 pts

Consider the following code snippet. Assume that this code snippet is executed and then answer the question below it.

```
movies = [
    {"title":  "A", "year":  19, "style":  "long", "genres":  ["g1", "g2"]},
    {"title":  "B", "year":  19, "style":  "short", "genres":  ["g2", "g3"]},
    {"title":  "C", "year":  20, "style":  "short", "genres":  ["g1", "g3"]},
    {"title":  "D", "year":  20, "style":  "long", "genres":  ["g1", "g2", "g3"]},
    {"title":  "E", "year":  20, "style":  "long", "genres":  ["g2"]}
]
```

What does the following evaluate to? movies[3]["genres"].index(movies[1]["genres"][1])

○ 2

○ 0

○ 1

○ 3

○ None

Question 21

1 pts

What is the output after the following code is executed?

```
import copy

grocery = {"Milk": 2,  "Eggs": 10}
shopping_cart_mom = [["Face Mask", "Hand Santitizer", "Thermometer"],
                     ["Lipsticks", "Face Cream"],
                     grocery]
shopping_cart_dad = shopping_cart_mom[:]
shopping_cart_dad[2]["Bread"] = 10
shopping_cart_dad[2]["Milk"] +=  1
print(shopping_cart_mom[2])
```

○ {"Milk": 3, "Eggs": 10, "Bread": 10}

○ ["Lipsticks", "Face Cream"]

○ {"Milk": 2, "Eggs": 10}

○ {"Milk": 2, "Eggs": 10, "Bread": 10}

Question 22

1 pts

Given the below code, which of the following options will print 3?

```
def alpha(obj_a, obj_B):
     if obj_B > obj_a:
          return obj_B + obj_a
     return obj_B

y = alpha
a = 1
b = 2
c = 3
d = 4
```

○ print(y(a,c) + y(b,d))

○ None of other answers -- they all lead to a TypeError

○ print(y(d,c) + y(b,a))

○ print(y(d,b) + y(c,a))

○ print(y(a,b) + y(c,d))

Question 23

1 pts

You are given the following code:

```
from collections import namedtuple
import math
Player = namedtuple("Player", ["name", "rating", "age"])
eve = Player("Evelyn", 87, 25)
kyl = Player("Kyle", 82, 24)
pam = Player("Pamela", 78, 26)
players = [pam, eve, kyl]
```

What is the value produced by running the following code?

```
players[2].age += 1
```

○ Kyle's age is set to 25.

○ Pamela's age is set to 27.

○ A new player object is created for Pamela.

○ A runtime error will occur.

Question 24

1 pts

What is the output after the following code is executed?

```
import copy
shopping_cart_mom = [["Face Mask",  "Hand Santitizer",  "Thermometer"],
                     ["Lipsticks",  "Face Cream"],
                     ["Milk",  "Eggs"]]
shopping_cart_dad = copy.copy(shopping_cart_mom)
shopping_cart_dad[1].pop(0)
shopping_cart_dad[1].pop(0)
shopping_cart_dad.append(["Keyboard",  "Earphone",  "PS4"])
print(shopping_cart_mom[1])
```

○ ["Lipsticks", "Face Cream"]

○ []

○ ["Keyboard", "Earphone", "PS4"]

○ ["Face Mask", "Hand Santitizer", "Thermometer"]

Question 25

1 pts

What is the output after the following code is executed?

```
shopping_cart_mom = ["Face Mask", "Hand Sanitizer", "Scarf"]
shopping_cart_dad = shopping_cart_mom
shopping_cart_dad.append("Thermometer")
shopping_cart_mom.remove("Scarf")
shopping_cart_dad.append("Oreo")
print(shopping_cart_mom)
```

○ ['Face Mask', 'Hand Sanitizer']

○ ['Face Mask', 'Hand Sanitizer', 'Thermometer', 'Oreo']

○ ['Thermometer', 'Oreo']

○ ['Scarf']

Question 26

1 pts

What will be printed and in what order after the code is run?

```
shopping_cart = [["Milk", 2.99], ["Eggs", 1.99],
                 ["Keyboard", 39.45], ["Monitor", 259.99]]
def receipt(items):
    total = 0
    for item in items:
        total += item[1]
        yield item[0] + ": $" + str(item[1])
    yield "Total: $" + str(total)
my_receipt = receipt(shopping_cart)
for item in my_receipt:
    print(item)
```

○

```
<generator object secret at 0x102ec2750>
```

○

```
Milk: $2.99
Eggs: $1.99
Keyboard: $39.45
Monitor: $259.99
Total: $304.42
```

○

```
Milk: $2.99
```

○

```
Milk: $2.99
Eggs: $1.99
Keyboard: $39.45
Monitor: $259.99
```

Question 27

1 pts

Replace the `???` in the following code to sort the `apartment_list` by the price.

```
apartment_list = [["GC 2b1b", 1140], ["Regent 2b1b", 799],
                  ["Saxony 2b1b", 745],  ["Lucky 2b1b", 1010]]
def price(apartment):
    return ???
apartment_list.sort(key=price)
apartment_list
```

○ price

○ apartment_list["price"]

○ apartment[1]

○ apartment[0]

○ apartment[2]

Question 28

1 pts

What are the types of greeting_fall and greeting, respectively?

```
def greeting_spring():
    return “Welcome to CS 220 :)!”

greeting_fall = greeting_spring
greeting = greeting_spring()
```

○ String, String

○ Function, String

○ String, Function

○ Function, Function

Question 29

1 pts

What is the output after the code is run?

```
def secret():
    yield "It's my secret."
    yield "I won't tell you!"
    yield "You will never know."
    yield "My mom doesn't know, either."

g = secret()
print(next(g))
```

○ "My mom doesn't know, either."

○ "It's my secret."

○ "You will never know."

○ This code does not run!

○ "I won't tell you!"

Question 30

1 pts

Which of the following descriptions is **not correct**?

○ When you open a file with mode 'w', if the file exists, the existing file is overwritten with the new file.

○ When you open a file whose path is the current working directory with mode "w", if the file does not exists, a new file is created.

○ Failure to close a file may lead to loss of changes to file.

○ When you open a file with a sub-directory in the path that does not exist, both the sub-directory and the file will be created.

○ You can have only limited number of open files.

Question 31

1 pts

Which of the following statements is **true** after we successfully run the following code? Please read all options carefully before answering!

```
import os
f = open('file.txt','w')
f.write('Rock ')
f.close()

os.mkdir('d1')
f = open(os.path.join("d1", "file.txt"),'w')
f.write('Paper ')
f.close()

f = open('file.txt','w')
f.write('Scissors ')
f.write('I won!')
f.close()

f = open('file.txt')
foo = f.read()
f.close()
```

○ ('d1' in os.listdir('.')) will return True

○ os.path.isfile(os.path.join("d1", "file.txt")) will return False

○ os.path.exists('file.txt') will return False

○ os.path.isdir('d1') will return False

○ The value stored in foo is "I won!"

Question 32

1 pts

What will be stored in data3 after the following code is executed?

```
f = open('file.txt','w')
f.write('alohomora')
f.close()

f = open('file.txt')
data1 = f.read()
data2 = data1.split('o')
f.close()

data3 = data1 [3]  + data2 [3]
```

○ 'hra'

○ 'om'

○ 'hr'

○ 'hh'

○ 'oo'

Question 33

1 pts

Which lines are executed in the following code? Use the marked line number (comment) to answer this question.

Hint: Note the difference between "which lines are executed" and "which lines are successfully executed". Question is asking "which lines are executed".

```
def f():
    a = 1      # line 1
    b = 0      # line 2
    return a+b # line 3
def g():
    try:
        c = 0                         # line 4
        print(f() / c)                # line 5
    except:
        print("An error happened!") # line 6
try:
    g()                         # line 7
except:
    print("An error happened!") # line 8
```

○ Lines 1, 2, and 3

○ Lines 1, 2, 3, 4, 6, and 7

○ Lines 1, 2, 3, 4, 5, 6, and 7

○ Lines 1, 2, 3, 4, 5, 6, 7, and 8

○ Lines 1, 2, 3, 4, 5, and 7

Question 34

1 pts

Which string will be stored in the variable "after" when the following code is completely executed?

```
before =  ['x','y','z']
after = ''
try:
    for i in range(-1,4):
        after += before [i]
except:
    after += after [0]
```

○ 'zxyzz'

○ 'xyzx'

○ 'zxyz'

○ 'yy'

○ 'xyz'

Question 35

1 pts

Which call to errorCatcher will **NOT cause an exception / error**?

```
def errorCatcher(x,y,z):
    assert type(x) == list and type(y) == list
    if x [-2]  > y [-1] :
        raise ArithmeticError("Incorrect numerical values")
    assert (x[1] - z) % 3 == 0
    return "Well done !"
```

○ errorCatcher([5,7],[2,8,6],1)

○ errorCatcher([3,2], [8,2], 0)

○ errorCatcher([4,2], [5], 7)

○ errorCatcher([-1,0], [1,2], -4)

○ errorCatcher([2,3,1], 4, -3)

Not saved Submit Quiz