25/35

CS 301 - Fall 2018

Instructors: Tyler Caraza-Harter and Adalbert Gerald Soosai Raj

Midterm Exam 2 - 10%

(Last) Surname: Lin (First) Given name: Ya-Yen

NetID (email): lin383 @ wisc.edu

IMPORTANT: Answers for all questions must be marked on a scantron. The answer marked on the scantron will be the only answer graded.

Fill in these fields (left to right) on the scantron form (use #2 pencil):

1. LAST NAME (surname) and FIRST NAME (given name), fill in bubbles
2. IDENTIFICATION NUMBER is your Campus ID number, fill in bubbles
3. Under ABC of SPECIAL CODES, write your lecture number and fill in:
   - **001** - MWF 8:50a (Tyler morning)
   - **002** - MWF 1:20p (Tyler afternoon)
   - **003** - TR 9:30a (Gerald)

You may only reference your notesheet. You may not use books, your neighbors, calculators, or other electronic devices on this exam. Please place your student ID face up on your desk. **Turn off and put away portable electronics now.**

When you're done, please hand in these sheets in addition to your filled-in scantron.

NUMBERED PAGES: 14

QUESTIONS: 35

INTENTIONALLY BLANK
(USE FOR ANY SCRATCH WORK)

Strings

1. Which of the following is True?
   a. `"bcd" < "abc"`
   b. `"BCD" < "abc"`
   c. `"bcd" < "ABC"`
   d. `"abbc" < "abb"`
   e. `"A2" < "A11"`

2. What does the following evaluate to?

```
"UID12345"[3:].lower().isdigit()
```

   a. True
   b. False
   c. 12345
   d. D12345
   e. None

3. What does the following code print?

```
s = "this is a test"
words = s.split(" ")
words[3] = "?" * len(words[-1])
print(" ".join(words))
```

   a. This is a test
   b. This is ???? test
   c. This is a ????
   d. This is a ?
   e. None of the above because an IndexError occurs

Lists

4. What will be printed and in what order?

```
nums = [10, 20, 30, 40, 50]
print(nums[-1])
print(nums[1:3])
```

   a. 40 and [20, 30]
   b. 50 and [20, 30, 40]
   c. 50 and [20, 30]
   d. IndexError: list index out of range

5. What is the output of the following code?

```
my_list = ["movies", "friends", "family", "books"]
my_list.append("music")
my_list.pop(3)
my_list.sort()
print(my_list)
```

   a. ['books', 'family', 'friends', 'movies',]
   b. ['books', 'family', 'friends', 'movies', 'music']
   c. ['movies', 'friends', 'family', 'books', 'music']
   d. ['family', 'friends', 'movies', 'music']

6. What will be printed and in what order?

```
nums = [1, 2, 3]
words = ["one", "two", "three"]
print(words[nums.index(2)])
words = ["zero"] + words
nums = [0] + nums
print(words[nums.index(2)])
```

   a. two, two
   b. two, one
   c. one, two
   d. two, three

Dictionaries

7. What will be printed and in what order?

```
d1 = {1 : "one", 2 : "two", 3 : "three"}
d1[0] = "zero"
d2 = dict()
for k in d1:
    d2[d1[k]] = k
print(d1[3])
print(d2["three"])
```

   a. three, KeyError
   b. three, 3
   c. 3, three
   d. KeyError, 3

8. What is the output of the following code?

```
words = ["one", "two", "three", "four", "five"]
d = {}
for word in words:
    if len(word) not in d:
        d[len(word)] = [word]
    else:
        d[len(word)] += [word]
print(d)
```

   a. {3: ['one', 'two'], 5: ['three'], 4: ['four', 'five']}
   b. {1: 'one', 2: 'two', 3: 'three', 4: 'four', 5: 'five'}
   c. {'one': 3, 'two': 3, 'three': 5, 'four': 4, 'five': 4}
   d. {3: ['two'], 5: ['three'], 4: ['five']}

9. Consider the following dictionary.

```
person = {}
person['name'] = 'Vinodh Balasubramanian'
person['address'] = {
    'street': '1, Gangai Amman Koil Street',
    'city': 'Chennai',
    'state': 'Tamil Nadu',
    'zip': 600014
}
```

How will you access the name of the city (i.e., Chennai) that this person lives in?

   a. person['city']
   b. person[1]['city']
   c. person['address'][1]
   d. person['address']['city']

Tuples

10. Which of the following is the correct way to the attribute x in Point p?

```
Point = namedtuple("Point", ['x', 'y'])
p = Point(10, 20)
```

   a. p{x}
   b. p("x")
   c. p[x]
   d. p["x"]
   e. p.x

11. Which of the following **cannot** be used as a key for a Python dictionary?
    a. tuples
    b. namedtuples
    c. lists
    d. strings
    e. negative integers

12. What is the output of the following code?

```
a = ([10], 30)
b = a
c = b[0]
c.append(20)
b = b + (40, 50)
print(a)
```

    a. ([10, 20], 30)
    b. ([10, 20], 30, 40, 50)
    c. ([10], 30, 40, 50)
    d. ([10], 30)

when concatenating two tuples (or lists), you're actually making a new tuple (or list)

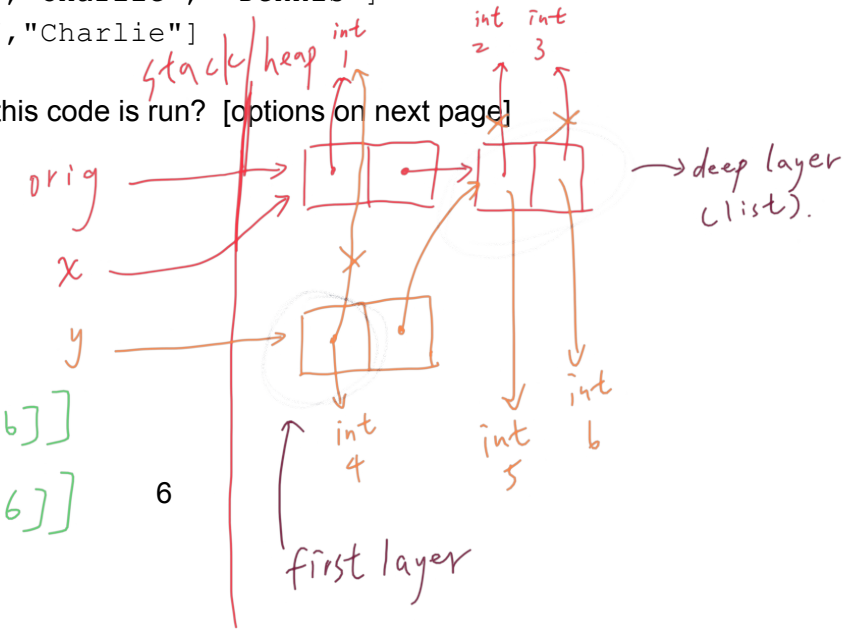
Copying

13. What is the value in **my_list** after this code is run?

```
my_list = ["Alice", "Bob", "Charlie", "Dennis"]
your_list = my_list
my_list.remove("Alice")
your_list.remove("Dennis")
print(my_list)
```

    a. ["Bob", "Charlie", "Dennis"]
    b. ["Bob", "Charlie"]
    c. ["Alice", "Bob","Charlie", "Dennis"]
    d. ["Alice", "Bob","Charlie"]

*when doing a shallow copy, only the first layer is truly a copy. every deeper layer is still tied to the original.



14. What is the value of x after this code is run? [options on next page]

```
orig = [1, [2, 3]]
x = orig
y = copy.copy(orig)
y[0] = 4
y[1][0] = 5
y[1][1] = 6
print(x)
```

a. [1, [2, 3]]
b. [4, [2, 3]]
c. [4, [5, 6]]
d. [1, [5, 6]]

15. What is the value of x after this code is run?

```
orig = [1,[2, 3]]
x = orig
y = copy.deepcopy(orig)
y[0] = 4
y[1][0] = 5
y[1][1] = 6
print(x)
```

a. [1, [2, 3]]
b. [4, [2, 3]]
c. [4, [5, 6]]
d. [1, [5, 6]]

Advanced functions

16. Using which command in a function body makes the function a generator function?
    a. yield
    b. return
    c. generate
    d. next

17. What will be printed and in what order?

```
def square_numbers(nums):
    for i in nums:
        yield (i*i)

my_nums = square_numbers([1,2,3,4,5])

for num in my_nums:
    print(num)
```

a. 1, 4, 9, 16, 25
b. <generator object square_numbers at 0x10c3877c8>
c. 1
d. 25

18. What is the output of the following code?

```
def foo(x):
    return x % 2 == 1

def special_sum(nums, check):
    total = 0
    for num in nums:
        if check(num):
            total += num
    return total
print(special_sum([1, 3, 4, 9], foo))
```

t t F t

a. 1
b. 4
c. 13
d. 17
e. None

File handling

19. Which of the following statements are true?
    a. When you open a file for reading, if the file does not exist, an error occurs
    b. When you open a file with mode “w”, if the file does not exist, a new file is created
    c. When you open a file with mode “w”, if the file exists, the existing file is overwritten with the new file
    d. All of the above

20. What is the type of data and lines in the following code?

```
f = open("file.txt")
data = f.read()
lines = data.split('\n')
```

read() method returns a string.
split() methods returns a list.

a. data is a string and lines is a list
b. data is a list and lines is a string
c. both data and lines are lists
d. both data and lines are strings

Nested Data Structures

21. What is printed?

```
data = [["alice", "in", "wonderland"], ["a","b","c"],
["hello"]]
print(data[2][0][-1])
```

a. a
b. hello
c. c
d. o
e. e

22. What is printed?

```
data = {1: {"f": "Alice"}, 2: {"f": "Bob", "l": "Smith"}}
print(data[1]["f"], data[1].get("l", "Doe"))
```

a. Alice
b. Alice Doe
c. Alice Smith
d. Bob Doe
e. Bob Smith

23. What is printed?

```
data = [["A", "B", "C"],[1,2,3], [4,5,6], [7,8,9]]
idx = data[0].index("B")
total = 0
for row in data[1:]:
    total += row[idx]
print(total)
```

a. 6
b. 12
c. 13
d. 15
e. 24

24. What is the output of the above snippet?

```
mapping = {}
for i in range(3):
    for j in range(2):
        mapping[(i,j)] = i + j

print(len(mapping), mapping[(1,1)])
```

a. 2 2
b. 3 1
c. 3 2
d. 6 1
e. 6 2

25. Consider the following code:

```
movies = [
  {"name": "m1", "actors": ["A", "B"], "genres": ["X"]},
  {"name": "m2", "actors": ["C", "D"], "genres": ["X", "Y"]},
  {"name": "m3", "actors": ["A", "C"], "genres": ["Y", "Z"]},
  {"name": "m4", "actors": ["B", "C"], "genres": ["X", "Z"]},
]

# key: actor name, val: list of unique genres actor has acted
in
actor_genres = {}
for movie in movies:
    for actor in movie["actors"]:
        for genre in movie["genres"]:
            if not actor in actor_genres:
                _____ <- what goes here?
            if not genre in actor_genres[actor]:
                actor_genres[actor].append(genre)
```

What should be inserted to populate `actor_genres` as described in the comment? (your code should not result in movies being changed).

a. `actor_genres[actor] = []`
b. `actor_genres[actor] = {}`
c. `actor_genres[actor] = movie["genres"]`
d. `actor_genres[genre] = []`
e. `pass`

26. What does the following code print?

```
mapping = {}
my_list = []

keys = [ 'k0', 'k1', 'k2' ]
for k in keys:
    mapping[k] = my_list

for i in range(3):
    k = keys[i]
    mapping[k].append(i)

print(mapping['k0'], mapping['k1'])
```

a. None None
b. [0] [1]
c. [1] [2]
d. [0, 1, 2] [0, 1, 2]
e. k0 k1

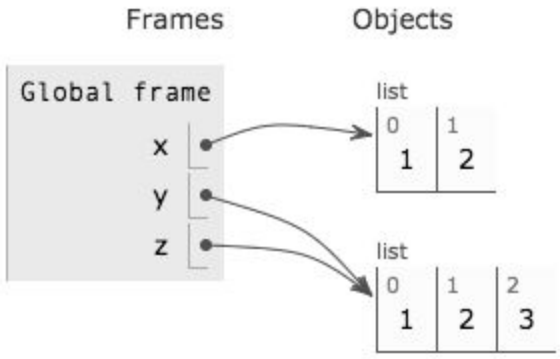
Objects and References

27. What does the following code print?

```
def foo(obj_a, obj_B):
      return obj_a + obj_b
y = foo
a = 1
b = 2
c = 3
print(y(a,b) + y(b,c))
```

a. 6
b. 8
c. 35
d. Code generates TypeError
e. Code prints representation of foo

28. Running which lines of code in Python Tutor would produce the following visualization?



a.
```
x = [1,2]
y = [1,2,3]
z = [1,2,3]
```
b.
```
x = [1,2,3]
y = x
z = x
x.pop(-1)
```
c.
```
x = [1,2]
y = x + [3]
z = y
```
d.
```
x = [1,2]
y = x
y.append(3)
z = y
```
e.
```
y = [1,2,3]
z = [1,2,3]
x = y[:2]
```

29. What does the following code print?

```
def foo(input_object):
    for item in input_object:
        item = item + 1
    return sum(input_object)

num_list = [1,2,3]
print(foo(num_list))
```

this for loop doesn't modify the content of the list.

a. 6
b. 9
c. [1, 2, 3]

d. [2, 3, 4]

Recursion

30. Just like infinite iteration, infinite recursion make a program run forever
    a. True
    b. False

31. What does `f("ABC")` return?

```
def f(text):
    if len(text) <= 1:
        return text
    return f(text[1:]) + text[0]
```

    a. A
    b. C
    c. ABC
    d. CBA
    e. None of the above because there is a stack overflow

32. What does `ternary(7)` return?

```
def ternary(n):
    if n < 3:
        return str(n)
    else:
        quo = n // 3
        rem = n % 3
        return ternary(quo) + ternary(rem)
```

    a. 3
    b. 7
    c. 12
    d. 21
    e. None of the above because there is a stack overflow

Error Handling

33. Which call to foo will **NOT** cause an AssertionError? [see options on next page]

```
def foo(x, y):
    assert type(x) == int and type(y) == int
    assert x % 2 == 0 and x > y
    return x - y
```

```
a. foo("4", 2)
b. foo(2, 2)
c. foo(3, 2)
d. foo(4, -3)
e. foo(4, 2.0)
```

34. What lines of code execute in the following?

```
a = 1      # line 1
b = 0      # line 2
c = a/b    # line 3
try:
     print("a/b is " + str(c))   # line 4
except:
     print("Got a divide-by-zero error!")   # line 5
print("exiting")   # line 6
```

a. Lines 1, 2, and 3
b. Lines 1, 2, 3, and 4
c. Lines 1, 2, 3, and 5
d. Lines 1, 2, 3, and 6
e. Lines 1, 2, 3, 4, 5, and 6

35. What does the following code print?

```
before = [10, 20]
after = []
for i in range(3):
     try:
          after.append(before[i])
     except:
          after.append(-1)
print(after)
```

a. [10, 20]
b. [10, 20, -1]
c. [10, 20, -1, -1]
d. [10, 20, 20]
e. [20, -1, -1]

Congratulations on finishing your midterm 2! :)