CS 301 - Fall 2018

Instructors: Tyler Caraza-Harter and Adalbert Gerald Soosai Raj

Final Exam — 20%

(Last) Surname: _____ (First) Given name: _____

NetID (email): _____ @wisc.edu

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, fill in bubbles:
   001 - MWF 8:50am (Tyler morning)
   002 - MWF 1:20pm (Tyler afternoon)
   003 - TR 10:00am (Gerald)
4. Under *F* of SPECIAL CODES, write **C** and fill in bubble **8**

There are multiple variants of this exam, so if you miss step 4 above (or do it wrong), the system won't grade you against the correct answer key, and your grade will be no better than if you were to randomly guess on each question. So don't forget!

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.

Use a #2 pencil to mark all answers. When you're done, please hand in these sheets in addition to your filled-in scantron.

(Blank Page)

1. What letters are printed, and in what order?

```
def get_letter(n):
   if n == 1:
       return 'X'
   else:
       return 'Y'
       return 'Z'

print(get_letter(3))
```

   A. X

   B. Y

   C. Z

   D. Y, Z

   E. X, Y, Z

2. What is printed?

```
import copy
stats = {}
results = []
for i in range(5):
    stats["score"] = 100+i
    results.append(copy.copy(stats))
print(results[2]["score"])
```

   A. 100

   B. 101

   C. 102

   D. 103

   E. 104

3. What is the output of the code snippet below?

```
def mystery(n):
    if n == 0:
        return 1
    elif n > 0:
        return 2 * mystery(n-1)
    else:
        return 1/mystery(-n)
print(mystery(-2))
```

A. 4

B. -4

C. 0.25

D. -0.25

E. RecursionError: maximum recursion depth exceeded

4. What is printed?

```
pt1 = (1, 2)
pt2 = (2, 3)
pt1[0] = 2
dist = ((pt2[0] - pt1[0]) ** 2 + (pt2[1] - pt1[1]) ** 2) ** (1/2)
print(dist)
```

A. 1.414

B. 1.0

C. 2.0

D. 6.0

E. TypeError: 'tuple' object does not support item assignment

5. What is printed? Assume file.txt DOES NOT exist before running this code snippet.

```
try:
    f = open("file.txt", "w")
    print("A")
    f.write("Hello")
    print("B")
except:
    print("C")
f.close()
```

   A. Nothing will be printed

   B. C

   C. A, B

   D. A, C

   E. A, B, C

6. What is printed and in what order?

```
def func1(items):
    return items[0]
def func2(items):
    items.sort(reverse=True)
    return items[0]
numbers = [4,5,3,2,1]
print(func1(numbers))
print(func2(numbers))
print(func1(numbers))
```

   A. 4, 1, 1

   B. 4, 1, 4

   C. 4, 5, 5

   D. 4, 5, 4

   E. 4, 1, 5

7. What is printed and in what order?

```
def hello():
    print("hi")

def goodbye():
    print("bye")

def multi_call(n, fn):
    if fn != None:
        for i in range(n):
            fn()

# careful, watch the parentheses!
multi_call(2, hello)
multi_call(3, goodbye())
```

A. hi, hi, bye

B. hi, hi, bye, bye, bye

C. hi, bye

D. hi, bye, bye, bye

E. Error: can't pass a function as an argument to another function

8. What numbers are printed, and in what order?

```
i = 1
while i < 10:
    print(i)
    if i % 3 == 0:
        continue
    i = i + 1
```

A. 1

B. 1, 2, 4, 5, 7, 8

C. 1, 2, 3, 4, 5, 6, 7, 8, 9

D. 3, 6, 9

E. the loops never stops running and keeps printing numbers

9. Assume the foo function header looks like this:

```
def foo(a, b, c=0):
```

Which call to foo must be INCORRECT?

A. `foo(1, 2)`

B. `foo(c=3, 1, 2)`

C. `foo(1, 2, 3)`

D. `foo(1, 2, None)`

E. `foo(1, c=3, b=2)`

10. What is printed?

```
y = 0
def cube_init(x):
    y = x ** 3
cube_init(-1)
print(y)
```

A. -8

B. -1

C. 0

D. 1

E. 8

11. What value does the following expression evaluate to?

```
(0 > 1 or 0 < 1) and (True or False) and not False
```

A. True

B. False

12. What is the smallest integer value for N that will cause "hi" to be printed?

```
if N > 1:
    if N > 2:
        if N > 3:
            if N > 4:
                pass
        print("hi")
```

A. 1

B. 2

C. 3

D. 4

E. 5

13. After running the following notebook cell, what will appear in the Out box?

```
def double(x):
    return x * 2

double(3)
double(7)
```

A. nothing

B. 6

C. 14

D. both 6 and 14 (on separate lines)

14. Assume the last cell in notebook is show below. The programmer clicks "Kernel" then "Restart and Run All" from the menu. She then manually executes this cell a second time. What is displayed in the Out box the second time?

```
nums = Series([1,2,3])
nums["total"] = sum(nums)
nums["total"]
```

A. nothing is displayed

B. 0

C. 6

D. 12

15. What are the values in b?

```
s = Series([9, 10, 11, 19, 20, 21])
b = s % 10 == 0
```

A. 10, 20

B. 9, 11, 19, 21

C. False, True, False, False, True, False

D. True, False, True, True, False, True

E. NaN, 10, NaN, NaN, 20, NaN

16. What are the value(s) in result?

```
s1 = Series([1, 2, 3])
s2 = Series([3, 2, 1])
s3 = Series([4, 5, 6])
result = s3[s1 != s2]
```

A. 2

B. 1, 3

C. 5

D. 4, 6

E. NaN, NaN

17. What is the general shape of this DataFrame?

```
df = DataFrame({
    "7": [1, 1, 1],
    "8": [4, 4, 4]
})
```

A. 1 column, 4 rows

B. 2 columns, 3 rows

C. 3 columns, 2 rows

D. 7 columns, 8 rows

E. 8 columns, 7 rows

18. What are the values in delta (in order from smallest index to largest)?

```
df = DataFrame({
    "A": [1, 2],
    "B": [2, 1],
    "x": [3, 7]
})

delta = df.set_index("A")["x"] - df.set_index("B")["x"]
```

A. 0, 0

B. -4, 4

C. 4, -4

D. 3, 7

E. 0, -4

19. Suppose a DataFrame (df) looks like this:

	A	**B**	**C**
X	1	2	3
Y	4	5	6
Z	7	8	9

What is the value of x after running the following?

```
subset = df[df["C"] < 7]
added = subset["A"] + subset["B"]
x = added.sum() - added.loc["Y"]
```

A. 3

B. 10

C. 15

D. Series([3, 3])

E. NaN

For the following SQL queries, assume the "flights" table is as follows:

source	**destination**	**cost**
MSN	ORD	400
MSN	MSP	350
ORD	MSN	400
MSP	ORD	200
MSP	MSN	325
MSP	MKE	150

20. What is the result of the following query?

```
SELECT MIN(cost) FROM flights
WHERE destination != 'MKE';
```

A. 150

B. 200

C. 400

D. MSP

E. ORD

21. Suppose you want to **filter** the results of a GROUP BY query based on some aggregate computed over each group. What kind of SQL clause provides this kind of filtering?

A. SELECT

B. HAVING

C. WHERE

D. LIMIT

22. Which is the only airport name that appears in the results?

```
SELECT source, COUNT() as total FROM flights
GROUP BY source
ORDER BY total DESC
LIMIT 1;
```

A. MKE

B. MSN

C. MSP

D. ORD

23. There is a clickable link in the following HTML. Where will clicking it take you?

```
<i>A.html</i><a href="B.html">C.html</a>
<img src="D.png">
```

A. A.html

B. B.html

C. C.html

D. D.html

24. What protocol does a client use to send a GET request for a resource to a server?

A. HTTP

B. HTML

C. CSV

D. SQL

25. Suppose the file at https://www.example.com/list.json contains this one line:

```
[77,88,99]
```

A Python program fetches the file with this code:

```
r = requests.get("https://www.example.com/list.json")
```

Which one expression will `NOT` produce 88? We don't care what type 88 is.

A. r.text[1]

B. json.loads(r.text)[1]

C. r.json()[1]

D. r.text.split(",")[1]

26. Suppose there is no resource with the URL https://www.example.com/missing.txt. Which letters will be printed by the following code?

```
try:
    r = requests.get("https://www.example.com/missing.txt")
    print("W")
    r.raise_for_status()
    print("X")
except:
    print("Y")
print("Z")
```

A. W, X, Y, Z

B. W, Y, Z

C. W, Z

D. Y, Z

27. You can use an index/key of -1 to lookup the last entry of which data structures?

A. list

B. dict

C. Series

D. all of the above

E. none of the above

28. What are the values in s (ordered from smallest to largest index)?

```
s = 4 + Series([1, 2, 3])
```

A. 4, 1, 2, 3

B. 1, 2, 3, 4

C. 5, 2, 3

D. 5, 6, 7

E. 5, NaN, NaN

29. What is printed

```
s = Series([1, 2, 3, 4], index=[3, 1, 4, 2])
print(s[1])
```

A. 0

B. 1

C. 2

D. 3

E. 4

30. In which scenario is using a log scale for the x-axis most natural?

A. different categories of trees appear along the x-axis

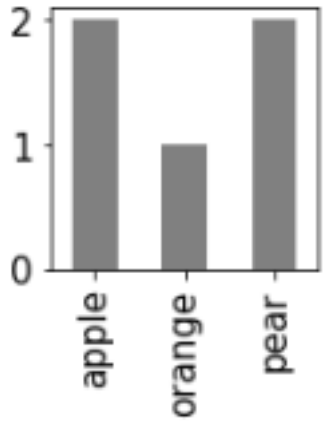
B. x values include both negative and positive numbers

C. x values include very small and very large positive numbers

D. there is only one y value for each x value

E. you want to show a stacked bar plot

31. Which snippet produces the following plot?



Assume the df DataFrame was produced as follows:

```
df = DataFrame({
    "fruit": ["apple", "apple", "pear", "pear", "orange"],
    "color": ["red", "green", "red", "green", "orange"],
    "count": [50, 25, 8, 10, 20]
})
```

A. df[“count”].plot.bar()

B. df.set_index(“fruit”)[“count”].plot.bar()

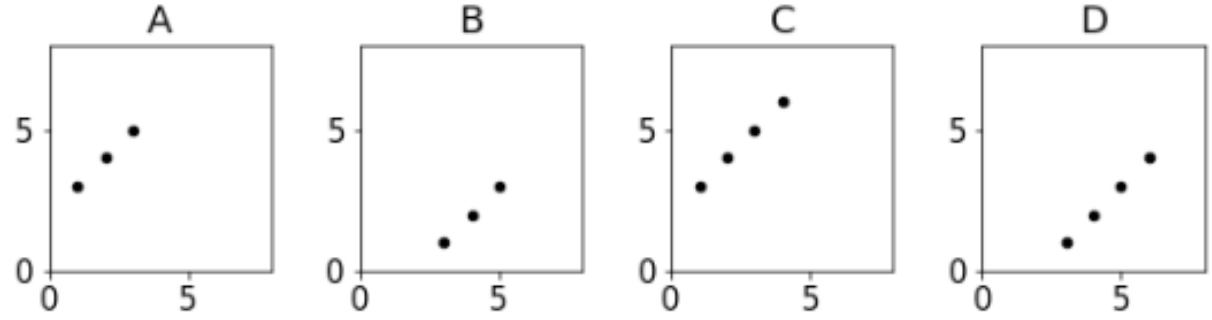
C. df.set_index(“fruit”)[“count”].sort_index().plot.bar()

D. df[“fruit”].value_counts().sort_values().plot.bar()

E. df[“fruit”].value_counts().sort_index().plot.bar()

32. Which plot corresponds to the following code snippet?

```
data = []
for i in range(3):
    data.append({"a": i + 1, "b": i + 3})
DataFrame(data).plot.scatter(x='a', y='b')
```



A. A
B. B
C. C
D. D

33. 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))
```

A. 6
B. 9
C. [1, 2, 3]
D. [2, 3, 4]

34. 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)

35. 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