CS 301 - Fall 2019

Instructor: Tyler Caraza-Harter

Final — 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 9:55am (Tyler morning)
   002 - MWF 4:35pm (Tyler afternoon)
4. Under *F* of SPECIAL CODES, write **6** and fill in bubbles **6**

If you miss step 4 above (or do it wrong), the system may not 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 have 110 minutes to take the exam. We reserve 10 minutes out of the 120 minute room reservation for handing out the exam materials. To compensate, the first six questions are quick True/False.

We won't always include imports in examples, but assume it has been done in a standard way (for example, assume `from pandas import DataFrame, Series` before each code example).

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.

Good luck! [feel free to do scratch work here]

Quick True/False

1. (T/F) If `N` is an odd int, then the following will be True. `N % 2 == 0`

   A. True **B. False**

2. (T/F) In a SQLite table, all the values in a column must be the same type.

   A. True B. False

3. (T/F) Python does NOT allow an entry in a list to refer back to the same list.

   A. True **B. False**

4. (T/F) namedtuples are immutable.

   A. True B. False

5. (T/F) the “i” in “iloc” stands for “index”.

   A. True **B. False**

6. (T/F) If both keyword and default arguments are available to fill a parameter, Python uses the keyword argument.

   A. True B. False

Functions

```
def counter(stuff):
    score = 0
    for something in stuff:
        if "o" in something:
            score += 5
        elif "i" in something:
            score += 3
        if "e" in something:
            score += 1
    return score

numbers = [1, 27, "42", 0, 42]
wines = ["Moscato", "Red Blend", "Chardonnay", "Pinot Noir", "Riesling"]

def search(myList, target):
    i = 0
    while True:
        if myList[i] == target:
            break
        i+= 1
    return i
```

7. What will `counter("i love Wisconsin weather!".split(" "))` return?

   A. 12 B. 14 **C. 15** D. 18 E. 22

8. What will `search(numbers, 42)` do? Be careful!

   A. loop forever B. crash with TypeError C. return 2 **D. return 3** E. return 4

9. What will `search(wines, "Pinot")` do? Be careful!

   A. loop forever **B. crash with IndexError** C. return 0 D. return 2 E. return 3

10. What is printed by the following?

```
total = 0
text = "12 3 45"
for c in text:
    if c.isdigit():
        total += int(c)
print(total)
```

   A. `0` B. `5` **C.** `15` D. `12 3 45` E. `51`

Databases

Consider the following code, assuming `Fifa19.csv` has the following columns:
Name, Age, Country, Overall, Club, Foot, Number

```
conn = sqlite3.connect("players.db") # line 1
players = pd.read_csv("Fifa19.csv")  # line 2
players.to_sql("players", conn, if_exists="replace", index=False) # line 3
```

11. The above creates a SQL table named "players"; which lines would we need to change if we wanted a different name for the SQL table? If there are multiple correct answers, choose the one involving the fewest changed lines.

    A. line 1 only B. line 2 only C. lines 2 and 3 only **D. line 3 only**

12. Which of the following is a valid query string (won't return an error)?

    A. `SELECT * FROM players WHERE Age > 40 ORDER BY Age DES`
    B. `SELECT FROM players WHERE Age > 40 ORDER BY Age DESC`
    C. `SELECT * FROM players WHERE Age > 40 ORDER BY DESC`
    D. `SELECT * FROM players WHERE Age > 40 ORDERBY Age DESC`
    E. `SELECT * FROM players WHERE Age > 40 ORDER BY Age DESC`

13. What does the following produce?

```
SELECT Country, AVG(Overall) AS rating
FROM players WHERE Foot = "Left"
GROUP BY Country ORDER BY rating DESC LIMIT 3
```

    A. all countries ordered by the average rating of their first 3 left-footed players
    B. the 3 countries with the highest average rating for their left-footed players
    C. the 3 countries with the lowest average rating for their left-footed players

14. Which line of Python code will produce the closest results to the following query?

```
SELECT Club, COUNT(*) as total
FROM players WHERE Country = "England"
GROUP BY Club ORDER BY total DESC
```

    A. `players["Club"]["England"].sum()`
    B. `players["England"]["Club"].value_counts()`
    C. `players["Country"]=="England"["Club"].value_counts()`
    D. `players[players["Country"]=="England"]["Club"].value_counts()`

matplotlib

15. What keyword argument can we pass to the `.plot.scatter(...)` method to guarantee the vertical axis starts at 0?

    A. y = 0 **B. ylim = 0** C. x = 0 D. xlim = 0

16. Assume we ran `ax = df.plot.scatter(x="A", y="B", ????)`. Note some parameters are hidden with `????`. How can we afterwards trigger an exception if values far to the right are cut off from the plot area?

    A. `raise df["A"].min() <= ax.get_xlim()[0]`
    B. `raise df.max() <= ax.get_xlim()`
    C. `raise Exception("x out of range")`
    D. `assert df["A"].max() <= ax.get_xlim()[1]`

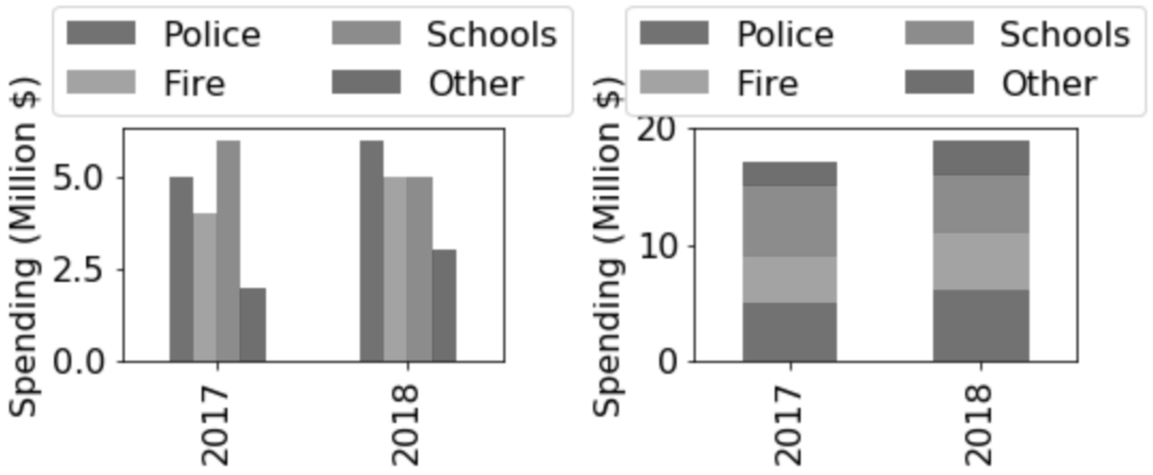
17. How can we increase the font size of all plots?

    A. `matplotlib.font = 16`
    B. `ax["font.size"] = 16`
    C. `matplotlib.rcParams["font.size"] = 16`

18. Which column of DataFrame `df` will be used for the x positions with the following call?
    `df.set_index("A")[["B", "C"]].plot.bar(stacked=True)`

    A. A B. B C. C D. bar E. stacked

19. Consider the two plots of the same data. Which makes it easier to see total spending per year across all areas?



    A. the plot with "clustered" bars **B. the plot with "stacked" bars**

HTTP and HTML

Assume the HTML for `http://example.com:312/amazon.html` is this:

```
<h1>amazon.html</h1><a href="product.html">review.html</a>
<ul><li>rating.html</ul>
```

Assume this code is being used to scrape that page (part of the code is hidden by `????`):

```
import requests
from bs4 import BeautifulSoup
try:
    r = requests.get("http://example.com:312/amazon.html")
    r.raise_for_status() # raises an HTTPError if page is missing
    doc = BeautifulSoup(????, "html.parser")
    link = doc.find("a").get_text()
except requests.exceptions.HTTPError as e:
    print("WARNING! Could not fetch page")
```

20. In URL `http://example.com:312/amazon.html`, what is `312`?

    A. a status code B. a domain name **C. a port number** D. a resource

21. What should replace `????` so that the code works?

    A. r **B. r.text** C. r.raise_for_status() D. r.json() E. r.dump()

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

    A. amazon.html **B. product.html** C. review.html D. rating.html

23. Assuming no exceptions, what is the ouput of `print(link)`?

    A. `<a href="product.html">review.html</a>`

    B. `"product.html"review.html`

    C. `product.htmlreview.html`

    D. `product.html`

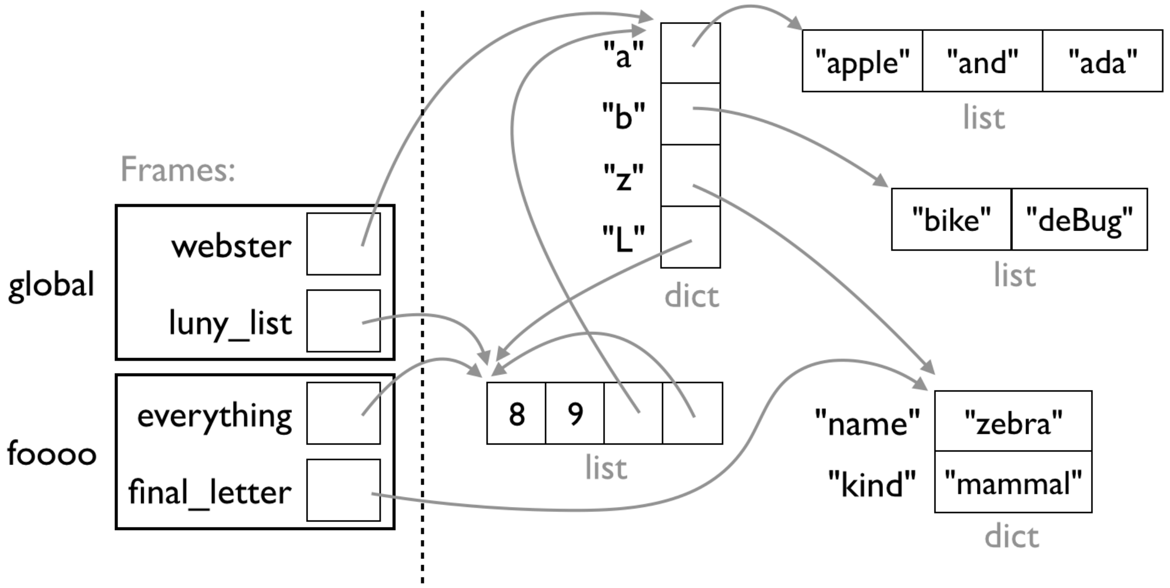
    E. `review.html`

24. Assuming no exceptions when the code runs, what HTTP method is used for the request, and what status code is returned in the HTTP response, respectively?

    A. POST, 200 B. POST, 404 **C. GET, 200** D. GET, 404

Data Structures

For the following, consult the following diagram of objects and references. Assume any code in the questions can access variables in both frames.



25. What does the following evaluate to? `"name" in final_letter`

    A. True B. False

26. `luny_list` refers to the same object as which of the following?

    A. `luny_list[3]` B. `luny_list[3][3]` C. `luny_list[3][3][3]` **D. all of the above**

27. Which of the following is True? Be careful!

    A. `webster["z"] is {"name":"zebra", "kind":"mammal"}`

    B. `webster["z"] == {"name":"zebra", "kind":"mammal"}`

    C. all of the above

    D. none of the above

28. What is the value of `everything[everything[1] - everything[0]]`?

    A. `0` B. `1` C. `8` **D.** `9`

29. What is the type of `everything[2]["L"][1]`?

    A. int B. str C. list D. dict

30. What does the following evaluate to? `str(luny_list[1]) * 3`

    A. `[3,3,3]` B. `9` C. `27` **D.** `999`

Pandas

```
s1 = Series([1,2,3],index=[3,1,2])
s2 = Series([1]) + s1
tbl = DataFrame({"x":[1,2,3], "y":[4,5,6]})
df = DataFrame({
  "country": ["Chile", "Bahrain", "Belize", "Kenya", "Marshall Islands"],
  "continent": ["S America", "Asia", "N America", "Africa", "Australia"],
  "coast": [5, 4, 3, 2, 1]
})
```

31. What are the values in `s2` (ordered from smallest to largest index)?

    A. 1,1,2,3 B. 2,3,4 C. 2,2,3 D. 2,NaN,NaN **E. NaN,NaN,NaN,NaN**

32. What is the ROWS x COLS size of `tbl`?

    A. 3x2 B. 2x3 C. 3x3 D. 1x6 E. 6x1

33. Which expression will evaluate to `"Marshall Islands"`?

    A. `df["country"].loc[-1]`
    B. `df["country"].iloc[-1]`
    C. `df.loc["country", -1]`
    D. `df.iloc["country", -1]`

34. What is the type of the following? `df["country"][4:]`

    A. string **B. Series** C. DataFrame D. list E. dict

35. What is the shortest way to correctly produce a result containing Chile?

    A. `df[df["continent"]=="S America" & df["coast"]==5]["country"]`
    B. `df[(df["continent"]=="S America") & (df["coast"]==5)]["country"]`
    C. `df[(df["continent"]=="S America") && (df["coast"]==5)]["country"]`
    D. `(df["continent"]=="S America") && (df["coast"]==5)`

Congrats on finishing CS 301 – have a refreshing winter break!

(Blank Page: you may tear this off for scratch work, but hand it in with your exam)

(Blank Page: you may tear this off for scratch work, but hand it in with your exam)