CS 220 - Fall 2021

Instructors: Meenakshi Syamkumar, Andrew Kuemmel

Final — 10%

(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 11:00am (Meena)
   002 - MWF 1:20pm (Meena)
   003 - MWF 8:50am (Andy)
   004 - MWF 9:55am (Andy)
4. Under *F* of SPECIAL CODES, write **A** and fill in bubble **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!

Many of the problems in this exam are related to the course projects, but some questions assume the availability of slightly different functions (e.g., for accessing the data). We won't have any trick questions where we call a function that doesn't exist and you need to notice. Thus, if you see a call to a function we haven't explicitly defined in the problem, assume the function was properly implemented (perhaps immediately before the code snippet we DO show) and is available to you.

You may only reference your note sheet. 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 (including smart watches) 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)

Review

1. Assume we have a function that tries to find the **maximum** value of a list as follows:

```
def find_max(somelist):
    max_value = 0
    for value in somelist:
        if value > max_value:
            max_value = value

    return max_value
```

What kind of list will guarantee this function returns the **wrong** result?

A. A list with all positive integers.

B. A list with at least 1 positive integer.

C. A list with at least 1 negative integer.

D. A list with all negative integers.

2. What is the value of `total` after executing the below code snippet?

```
total = 0
for i in range(10):
    if i % 2 == 0:
        continue
    elif i > 6:
        break
    total += i
```

A. 0 B. 1 **C. 9** D. 16 E. 21

3. Consider the below function definition.

```
def adjacent(s = "abc"):
    for i in range(len(s)):
        if s[i] == s[i + 1]:
            return True
    return False
```

What would `adjacent()` evaluate to?

A. `False` B. `True` C. Syntax error **D. Runtime error**

4. What is the output of the following code snippet?

```
i = 16
while i > 1:
    if i > 10:
        print("a", end = "")
    elif i > 5:
        print("b", end = "")
    if i > 2:
        print("d", end = "")
    i /= 2
```

A. `abdd` **B.** `adbdd` C. `adbd` D. `abd`

5. In the below code, how many times will the `while` condition get evaluated?

```
i = 16
while i > 1:
    if i > 10:
        print("a", end = "")
    elif i > 5:
        print("b", end = "")
    if i > 2:
        print("d", end = "")
    i /= 2
```

A. 4 **B. 5** C. 6 D. 7

6. What will be printed after this code runs?

```
a = [1, 2]
b = [3]
c = b
c.append(1)
c.sort()
a.append(b)
print(a)
```

A. `[1, 2, [3, 1]]`

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

C. `[1, 2, 1, 3]`

D. `[1, 2, 3]`

E. `[1, 2, [1, 3]]`

7. What will be printed after this code runs?

```
s = "I love cs220"
words = s.split(" ")
p = words.pop(1)
print(p)
```

A. `I`

B. `love`

C. `["love", "cs220"]`

D. `["I", "cs220"]`

E. `["I", "love", "cs220"]`

8. Which of the following will correctly sort players by name?

```
players = [{"name":"Cindy", "jersey":4, "age":30},
           {"name":"Alice", "jersey":20, "age":25},
           {"name":"Bob", "jersey":14, "age":32}]
```

A. `sorted(players["name"])`

B. `sorted(players, key=name)`

C. `sorted(players.items(), key=name)`

D. `sorted(players, key=lambda p:p["name"])`

E. `sorted(players.items(), key=lambda p:p["name"])`

9. Which of the following will generate a dictionary with name to jersey mapping?

```
players = [{"name":"Cindy", "jersey":4, "age":30},
           {"name":"Alice", "jersey":20, "age":25},
           {"name":"Bob", "jersey":14, "age":32}]
```

A. `{k:v for (k, players["jersey"]) in players}`

B. `{d["name"]:d["jersey"] for d in players}`

C. `{d[0]:d[1] for d in players}`

D. `{k:v for (k, v) in players.items()}`

Hurricanes (Database)

Consider the following code, and the corresponding output:

```
import sqlite3
import pandas as pd

conn = sqlite3.connect("hurricane.db")
hurricanes = pd.read_sql("select * from hurricanes", conn)
hurricanes
```

	Name	Formed	Speed	Damage	Deaths
0	Baker	1950	105	2.55	38
1	Camille	1969	175	1420.00	259
2	Eloise	1950	125	560.00	80
3	Frederic	1969	130	1770.00	12
4	Elena	1985	125	1300.00	9

10. What SQL query best translates the below Pandas code?

```
hurricanes[hurricanes["Speed"] > 120]["Damage"].mean()
```

A. `select avg(damage) from hurricanes if speed >= 120`

B. `query damage from hurricanes if speed > 120`

C. `select damage.mean() from hurricanes where speed > 120`

D. `select avg(damage) from hurricanes where speed > 120`

11. Which SQL query will return the name of the deadliest hurricane? Deadliest hurricane is the one with the most deaths.

A. `SELECT name FROM hurricanes ORDER BY deaths LIMIT 1`

B. `SELECT name FROM hurricanes ORDER BY deaths DESC LIMIT 1`

C. `QUERY name FROM hurricanes ORDER BY deaths LIMIT 1`

D. `SELECT * FROM hurricanes LIMIT 1`

12. Which SQL query answers the following question: Since 1960, which years have had hurricanes with a total damage greater than 3000?

    A. `SELECT formed, SUM(damage) as total FROM hurricanes WHERE formed >= 1960 AND total > 3000 GROUP BY formed`

    B. `SELECT formed, SUM(damage) as total FROM hurricanes WHERE total >= 1960 GROUP BY formed HAVING formed >= 1960`

    C. `SELECT formed, SUM(damage) as total FROM hurricanes WHERE formed >= 1960 GROUP BY formed HAVING total > 3000`

    D. `SELECT formed, damage.SUM() as total FROM hurricanes GROUP BY formed HAVING formed >= 1960 AND total > 3000`

13. Which hurricane will get excluded when we run the following SQL query?

    ```
    select name from hurricanes order by formed asc limit 4
    ```

    A. Baker B. Camille C. Eloise D. Frederic **E. Elena**

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

    ```
    SELECT Formed, COUNT(*) as count
    FROM hurricanes WHERE Damage > 1000
    GROUP BY Formed ORDER BY count DESC
    ```

    A. `hurricanes["Formed"]["Damage" > 1000].sum()`

    B. `hurricanes[hurricanes["Damage"] > 1000]["Formed"].value_counts()`

    C. `hurricanes["Damage" > 1000]["Formed"].value_counts()`

    D. `hurricanes["Damage"]>1000["Formed"].value_counts()`

15. How many rows will the following SQL query return?

    ```
    select name from hurricanes where deaths < 80 limit 4
    ```

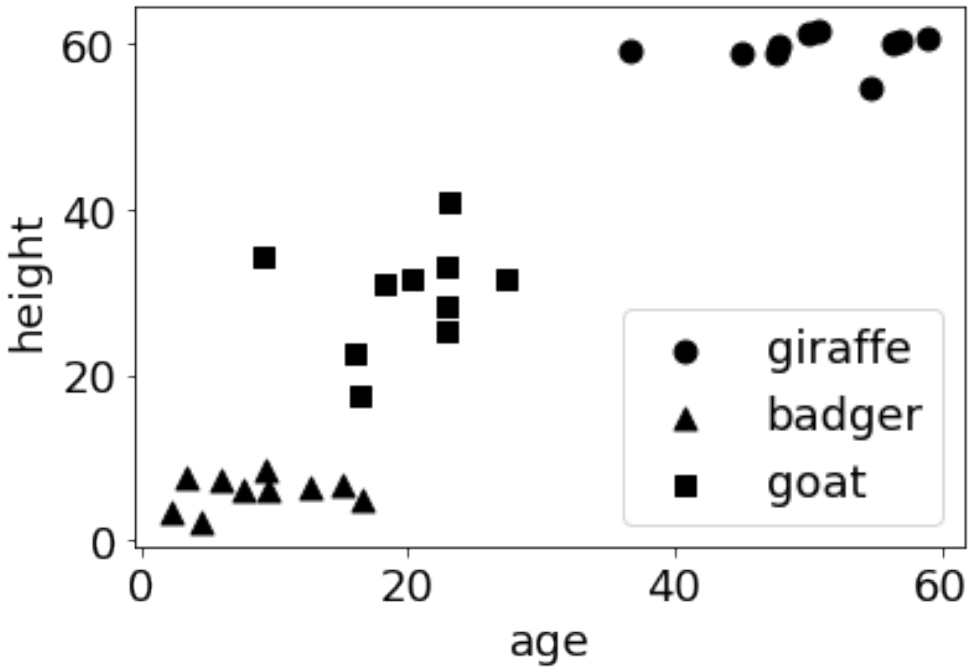
    A. 0 B. 1 C. 2 **D. 3** E. 4

Plotting

16. Assume the following:

    - `animal` DataFrame contains the columns `name`, `age`, and `height`
    - unique values stored within `name` column are badger, goat, and giraffe

    What are the arguments we need to pass for `scatter` function to produce the plot?




```
plot_area = None
markers = ["o", "^", "s"]
unique_animals = set(animal["name"])
for curr_animal in unique_animals:
    sub_df = animal[animal["name"] == curr_animal]
    plot_area = sub_df.plot.scatter(???, ???, ???,
                                    marker=markers.pop(0),
                                    label=curr_animal)
```

A. `x="age", y="height", ax=plot_area`

B. `x="height", y="age", ax=plot_area`

C. `x="height", y="age", ax=ax`

D. `x="age", y="height", ax=ax`

E. `x="age", y="height", ax=None`

17. How can we set figure size of a plot?

    A. setting `matplotlib.rcParams["font.size"] = 16`

    B. passing named keyword argument as `size = (width, height)`

    C. passing named keyword argument as `figsize = (width, height)`

    D. setting `matplotlib.rcParams["font.figsize"] = (width, height)`

18. Passing `True` to which of the following parameter enables us to set logarithmic scale for y-axis?

    A. `logx` **B.** `logy` C. `log` D. `sharey` E. `sharex`

19. For **every** x-axis tick label, how many bars will get produced when we execute the below code?

```
df.set_index("A")[["B", "C", "D", "E"]].plot.bar(stacked=True)
```

    A. 1 B. 2 C. 3 D. 4 E. 5

20. Suppose `s` is a Series, when we invoke `s.plot.bar()`, which of the following in the Series represent the x-axis?

    A. values **B. index** C. integer position D. columns

21. Which of the following Series would produce a misleading line plot?

    A. `Series([4, 5, 5, 6], index = [1, 2, 3, 4])`

    B. `Series([0, 1, 3, 2, 4], index = [0, 2, 7, 5, 1])`

    C. `Series([7, 5, 4, 3], index = [4, 5, 1, 2]).sort_index()`

    D. `Series([6, 4, 5, 3, 1, 2])`

Vaccinations (Pandas)

Consider the following code:

```
from pandas import Series, DataFrame

vacc = DataFrame({
    "Date": ["10/11/21", "10/12/21", "10/13/21", "10/14/21"],
    "Croatia": [173, 174, 175, 176],
    "France": [4513, 4518, 4525, 4530],
    "Lebanon": [136, 134, 139, 140],
    "Suriname": [18, 19, 21, 42]
})
```

22. Which of the following will **NOT** evaluate to 4518?

    A. `vacc["France"][1]`

    B. `vacc[1]["France"]`

    C. `vacc["France"].iloc[1]`

    D. `vacc.loc[1, "France"]`

    E. `vacc.iloc[1, 2]`

23. What is the type of `vacc["Lebanon"]`?

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

24. What is the ROWS x COLS dimension of the DataFrame `vacc`?

    A. 4 x 5 B. 5 x 4 C. 1 x 5 D. 5 x 1 E. 4 x 4

25. What are the values in the Series produced by the below expression? Assume that the values are ordered based on index.

```
vacc["Suriname"] + (-1)
```

    A. `19, 20, 22, 43`

    B. `-18, -19, -21, -42`

    C. `17, 18, 20, 41`

    D. `18, 19, 21, 42, -1`

26. Which of the following expressions answers the following question: What are the dates on which Lebanon and Suriname have had more than 135 and 20 vaccinations respectively?

A. `vacc[(vacc["Lebanon"] > 135) and (vacc["Suriname"] > 20)]["Date"]`

B. `vacc[(vacc["Lebanon"] > 135) & (vacc["Suriname"] > 20)]["Date"]`

C. `vacc[vacc["Lebanon"] > 135 & vacc["Suriname"] > 20]["Date"]`

D. `[(vacc["Lebanon"] > 135) & (vacc["Suriname"] > 20)]["Date"]`

27. What are the values in the Series that gets created with the following line of code? Assume that the values are ordered based on index.

```
vacc["Croatia"] + Series([2, 5], index=[1, 2])
```

A. 173, 176, 180, 176

B. 173, 174, 175,176

C. NaN, NaN, NaN, NaN

D. NaN, 176, 180, NaN

28. Which of the following best represents the output of the below expression?

```
vacc.iloc[0][1:3]
```

A. `Series({1:"10/12/21", 2:"10/13/21"})`

B. `Series({1:"10/12/21", 2:"10/13/21", 3:"10/14/21"})`

C. `Series({"Croatia":173, "France":4513})`

D. `Series({"Croatia":173, "France":4513, "Lebanon":136})`

29. Which expression answers this question: On what date did France have maximum vaccinations?

A. `vacc[vacc["France"] == vacc["France"].max()]["Date"].iloc[0]`

B. `vacc["France"] == vacc["France"].max()["Date"].iloc[0]`

C. `vacc[vacc["France"].max()]["Date"].iloc[0]`

D. `vacc["France"] == vacc["France"].max().iloc[0]`

Web

Assume that these are the contents of the file "http://somewhere.com/shopping.html":

```
<ul>
    <li><a href="greenbush.html">I want <b>donuts</b></a>
</ul>
```

Assume this code is being used to create a local copy of shopping.html (part of the code is hidden by ????):

```
import requests

def download(url):
    try:
        r = requests.get(url)
        ????
        f = open("shopping.html", 'w')
        f.write(r.text)
        f.close()
    except requests.exceptions.HTTPError as e:
        print("WARNING! Could not fetch page")

download("http://somewhere.com/shopping.html")
```

30. What should replace ???? so that the local file shopping.html would only contain the contents of remote file shopping.html, if HTTP request is successful?

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

31. In URL http://somewhere.com/shopping.html, what is somewhere.com?

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

    Consider the below code snippet that is trying to parse the content of local file shopping.html:

```
from bs4 import BeautifulSoup

if os.path.exists("shopping.html"):
    f = open("shopping.html")
    html = f.read()
    doc = BeautifulSoup(????, "html.parser")
    link = doc.find("a")
    f.close()
```

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

A. `r.text` B. `f` C. `r.json()` **D.** `html` E. `html.text`

33. What is the output of `print(link.get_text())`?

A. `greenbush.html`

B. `I want donuts`

C. `<a href="greenbush.html">I want <b>donuts</b></a>`

D. `["greenbush.html", "I want donuts"]`

34. What is the type of `link.attrs`?

A. list B. tuple **C. dict** D. set E. Series

35. Which HTML tag in `shopping.html` does not have a closing tag?

A. `ul` **B.** `li` C. `a` D. `b`

(Blank Page)

(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)