(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 08:50 AM (Mike)
   - 002 - MWF 11:00 AM (Mike)
   - 003 - MWF 09:55 AM (Louis)
   - 004 - MWF 01:20 PM (Louis)
4. Under *F* of SPECIAL CODES, write **C** and fill in bubble **8**

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 this exam and your note sheet in addition to your filled-in scantron.

Rankings Pandas

In P12, we analysed World University Rankings data. Here, only consider the first 6 lines of data and use the data to answer the following questions:

```
rankings = pd.read_json('rankings.json').iloc[:6]
rankings
```

–	Year	World Rank	Institution	Country	National Rank	Educ Rank	Employ Rank	Fac Rank	Res Rank	Score
0	2021	1	Harvard Univ.	USA	1	1.0	1.0	1.0	1.0	100.0
1	2021	2	MIT	USA	2	4.0	12.0	2.0	8.0	96.7
2	2021	3	Stanford Univ.	USA	3	10.0	4.0	3.0	2.0	95.1
3	2021	4	Univ. of Cambr.	UK	1	3.0	25.0	4.0	10.0	94.1
4	2021	5	Univ. of Oxford	UK	2	7.0	27.0	9.0	4.0	93.3
5	2021	6	Princeton Univ.	USA	4	5.0	15.0	7.0	70.0	92.6

1. What is the data type of `rankings["Country"]`?

    A. DataFrame

    B. str

    C. list

    D. Series

    E. int

2. Which of the following statements will create a `list` to display the institutions which have a have better `National Rank` than `World Rank`. Note 1 is the best rank.

    A. `list(rankings[rankings['National Rank'] - rankings['World Rank'] >= 0]['Institution'])`

    B. `rankings[rankings['National Rank'] - rankings['World Rank'] <= 0]['Institution']`

    C. `list(rankings[rankings['National Rank'] - rankings['World Rank'] < 0]['Institution'])`

    D. `list(rankings[rankings['National Rank'] - rankings['World Rank'] > 0]['Institution'])`

3. What is the output of the following code?

```
list(rankings['Score'].iloc[3:])
```

      A. `[94.1, 93.3, 92.6]`

      B. `[95.1, 94.1, 93.3, 92.6]`

      C. `[100.0, 96.7, 95.1]`

      D. `[96.7, 95.1, 94.1, 93.3, 92.6]`

      E. `[93.3, 92.6]`

4. Which of the following code snippets will find the largest National Rank?

      A. `rankings['National Rank'].max()`

      B. `rankings['National Rank'].sort_values().iloc[0]`

      C. `rankings['National Rank'].sort_values(ascending = False).iloc[0]`

      D. Both A and B

      E. Both A and C

Rankings SQL

5. For the next two questions, assume that the database version of rankings has been prepared and the function `query` is defined and will correctly execute an SQL query on the database when given a string representing an SQL query as input.

```
rankings = pd.read_json("rankings.json")
conn = sqlite3.connect("rankings.db")
rankings.to_sql("rankings", conn, if_exists="replace", index=False)
def query(qry):
    return pd.read_sql(qry, conn)
```

Consider the code below. Which Pandas command will produce the same DataFrame?

```
query("""
SELECT 'Country', AVG('Res Rank')
FROM rankings
WHERE 'Year' = 2023
GROUP BY 'Country'
ORDER BY AVG('Res Rank')
LIMIT 10
""")
```

    A. `rankings[rankings["year"]==2023][["Country","Res Rank"]].groupby("Country").mean().sort_values("Res Rank").head(10)`

    B. `rankings[rankings["year"]==2023][["Country","Res Rank"]].groupby("Country").mean().sort_values("Res Rank").iloc[:10]`

    C. `rankings[["Country","Res Rank"]][rankings["year"]==2023].groupby("Country").mean().sort_values("Res Rank").iloc[:10]`

    D. `All of the above`

6. Which SQL query will produce the same result as the following pandas statements.

```
usa = rankings[rankings["Country"]=="USA"]
usa = usa.sort_values("Score",ascending=False)[["Institution","National Rank"]]
usa = usa.iloc[:5]
```

  A. query("""select 'Institution', 'National Rank' from rankings
    where 'Country'="USA" order by 'Score' desc limit 5""")

  B. query("""select 'Institution', 'National Rank' from rankings
    group by 'Country' having 'Country'="USA"
    order by 'Score' desc limit 5""")

  C. query("""select 'Institution', SORTED('National Rank')
    from rankings
    group by 'Country' having 'Country'="USA"
    limit 5""")

  D. query("""select 'Country' = USA from rankings
    order by 'Score' desc
    columns 'Institution', 'National Rank'
    limit 5""")

General Part 1

7. Consider the pseudocode below, assuming that the variable `arr` is already defined and contains integer values.

```
1. Sort the list 'arr' in ascending order
2. Initialize 'm' to None, 'max' to 0, and 'current' to 1
3. Iterate through each index 'i' in 'arr' and for each index,
   perform steps 4 and 5. (Note if 'arr' contains 5 numbers, 'i'
   will begin at 0 and end at 4)
4. If 'i' is greater than 0 AND the
   value in the list at index 'i' is equal to the value
   in the list at index 'i-1', increment 'current'. Otherwise,
   set 'current' to be 1.
5. If 'current' is greater than 'max', set 'm' to be the value in
   the list at index 'i' and set 'max' to be equal to 'current'
```

What will be the value stored in the variable `m` after the pseudocode completes?

    A. The average of the list

    B. The mode of the list

    C. The median of the list

    D. The maximum of the list

    E. None of the above

8. What will be the output of the code below?

```
print(712 // 10 % 10)
```

    A. 1

    B. 2

    C. 7

    D. 712

    E. ZeroDivisionError

9. What will be the total number of characters printed out on LINE 1 and LINE 2, respectively?

```python
print("Alice" + ": " + " " * (9 - len("Alice")) + "|" * 10) # LINE 1
print("Bob" + ": " + " " * (9 - len("Bob")) + "|" * 8)       # LINE 2
```

   A. 12, 10
   B. 17, 19
   C. 19, 17
   D. 21, 19
   E. 26, 24

10. What will be the output of the code below?

```python
def add4(w = 0, x = 0, y = 0, z = 0):
    return w + x + y + x

print(add4(10, 25, 42, x = 9))
```

   A. 0
   B. 61
   C. 77
   D. 86
   E. An error

11. Which of the answer choices when substituted for the ??? in the code below will make `func1` and `func2` logically equivalent? Assume A and B are passed Boolean values.

```python
def func1(A, B):
    if A == False:
        return True
    else:
        return B

def func2(A, B):
    ???
```

   A. `return A or not B`
   B. `return not A and B`
   C. `return not A and not B`
   D. `return not A or B`
   E. `return A and B`

12. Assume that a gambler wants to place the following bets:
    45 8 10 5 25
    They use the following code to place their bets:

```
chips = 50
round_num = 0
while chips > 0 and round_num < 5:
    wager = int(input("New round. How much do you want to wager? "))

    while wager < 0 or wager > chips:
        wager = int(input("Bad input. How much do you want to wager? "))

    chips += -1 * wager
    print("You now have {} chips".format(chips)) # LINE 10
    round_num += 1

print("Thanks for playing, you now have {} chips".format(chips))
```

How many times will LINE 10 be evaluated?

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

General Part 2

13. What will be the output of the following code:

```
values = [10,10,20,40,40,60,70]
my_set = set(values)
popped = my_set.pop()
print(popped)
```

      A. `10`

      B. `60`

      C. `70`

      D. `A random value from the set`

      E. `The code will raise an error`

14. What will be the output of the following code:

```
nums_dict = {1:"one", 2:"two", 3:"three"}
nums_dict.pop(1)
print(nums_dict)
```

      A. `{1:  "one"}`

      B. `{2:  "two", 3:  "three"}`

      C. `{1:  "one", 2:  "two", 3:  "three"}`

      D. `None`

      E. `The code will raise an error`

15. Assume the file data.json is in the current directory and contains the following contents:

```
{
    "uno" : "one",
    "dos" : "two",
    "tres" : "three"
}
```

In addition, the following code has been provided to you:

```
import json
def read_json(path):
    with open(path, encoding="utf-8") as f:
        return json.load(f)

def write_json(path, data):
    with open(path, 'w', encoding="utf-8") as f:
        json.dump(data, f, indent=2)
```

Then, what will be the output of the code below:

```
sp_eng_dict = read_json("data.json")
sp_eng_dict["cuatro"] = "four"
write_json("data.json", sp_eng_dict)
new_dict = read_json("data.json")
print(len(new_dict))
```

      A. 3

      B. 4

      C. 7

      D. `read_json` will throw an error

      E. `write_json` will throw an error

16. What will be the output of the following code:

```python
x = [["a","b","c"], ["d","e","f"]]
y = x[:]
y[0][0] = "yyy"
x[0][0] = "zzz"
print(y[0][0])
```

    A. yyy

    B. zzz

    C. a

    D. The code will raise an error

17. What will be the output of the following code:

```python
def func(f, n):
    for i in range(n):
        f()

def hammer():
    print("Bang", end=" ")

def screwdriver():
    print("Whrr", end=" ")

func(hammer, 2)
func(screwdriver, 3)
```

    A. Bang Bang Whrr Whrr Whrr

    B. Bang Whrr Bang Whrr Whrr

    C. This code will raise an error because functions are called before their definitions

    D. This code will raise an error because functions cannot be passed as arguments

18. What will be the output of the following code:

```
user_type = "Manager"

try:
    if user_type != "Admin":
        raise PermissionError("User does not have permission")
        print("Access denied!")
    else:
        print("Access granted")
except Exception as e:
    print("Error:", e)
```

    A. `Access denied!`

    B. `Access granted`

    C. `User does not have permission`

    D. `PermissionError("User does not have permission")`

    E. `Error:  User does not have permission`

General Part 3

19. What will be the output of the following code?

```
import pandas as pd
nums = pd.Series([11, 12, 15, 17])
print(list(nums % 3==0))
```

      A. `[False, True, True, False]`

      B. `[12, 15]`

      C. `[True, False, False, True]`

      D. `[11, 17]`

      E. `None of the above`

20. Consider the following code:

```
df = pd.DataFrame({"Year":[2000,2001,2002,2003], "Income":[500, 200, 1000, 350]})
df = df[???]
```

We want to only keep the rows of df which have a Year of at least 2001 and an Income less than 500. Which of the following code snippets could replace the ??? to accomplish this?

      A. `(df["Year"]>=2001 & df["Income"]<500)`

      B. `(df["Year"]>=2001 and df["Income"]<500)`

      C. `(df["Year"]>=2001) & (df["Income"]<500)`

      D. `(df["Year"]>=2001) and (df["Income"]<500)`

      E. `All of the above`

21. Consider the following code. Assume 'file.json' exists and can be found by the code in the current directory.

```
path = "file.txt"
text = "Hello"
f = open(???, ???, encoding="utf-8")
f.write(???)
???
```

We want the code to write 'Hello' to file.txt and properly close the file. Which group of code snippet(s) could replace the ???'s in order to accomplish this?

    A. `text, "w", path, f.close`

    B. `text, "r", path, f.close()`

    C. `path, "w", text, f.close`

    D. `path, w, text, f.close()`

    E. `path, "w", text, f.close()`

22. Consider a database FH of fire hydrants, with two columns: (1) color, and (2) year. These represent the color and the year of construction of each fire hydrant. Which qry should we pass to pd.read_sql to find how many hydrants of each color we placed in the 21st century?

    A.
```
qry = """
SELECT COUNT(*) FROM FH
WHERE year >= 2000
"""
```

    B.
```
qry = """
SELECT color, COUNT(*) FROM FH
WHERE year >= 2000
"""
```

    C.
```
qry = """
SELECT color, COUNT(*) FROM FH
WHERE year >= 2000
GROUP BY color
"""
```

    D.
```
qry = """
SELECT color, COUNT(*) AS count FROM FH
WHERE year >= 2000
GROUP BY color
HAVING count > 1
"""
```

23. Consider a dataframe df with information for a class of students, including name, phone, email, and date of birth. A student is missing if their name is missing, or if both their phone and email are missing. How can we find how many students are missing?

    A. `((df["name"].isna()) & (df["email"].isna()) &`
       `(df["phone"].isna())).value_counts()`

    B. `((df["name"].isna()) | (df["email"].isna()) |`
       `(df["phone"].isna())).value_counts()`

    C. `((df["name"].isna()) | ((df["email"].isna()) |`
       `(df["phone"].isna()))).value_counts()`

    D. `((df["name"].isna()) | ((df["email"].isna()) &`
       `(df["phone"].isna()))).value_counts()`

24. Consider a Dataframe df of fire hydrants, with two columns: (1) color, and (2) year that represent the color and the year of construction of each fire hydrant. How can we draw a horizontal bar plot of the number of hydrants of each color?

    A. `df.plot.barh(x="color", y="count")`

    B. `df.plot.line(x="color", y="count")`

    C. `df["color"].value_counts().plot.bar()`

    D. `df["color"].value_counts().plot.bar(horizontal = True)`

    E. `None of the above`

25. Which of the following is true regarding the code below?

`df.plot.line(x="Time", y="Temperature", color="b", linewidth=2)`

    A. `help(pd.DataFrame.plot.line) may offer assistance with the color`
       `option`

    B. `Data points along the x-axis represent values from the "Time"`
       `column in the DataFrame df.`

    C. `Data points along the y-axis represent values from the`
       `"Temperature" column in the DataFrame df.`

    D. `The "linewidth" argument specifies the width of the line in the`
       `plot.`

    E. `All of the above`

26. Assume that the url "https://www.cs220.com/vacation.html" exists and will respond to requests by returning an HTML file. Which of the options below could replace the ??? in the code to successfully save the HTML code to a file on the user's machine?

```
import requests
from bs4 import BeautifulSoup

def download(url):
    try:
        r = requests.get(url)
        f = open("travel.html", "w", encoding="utf-8")
        f.write(???)
        print("Success!")
    except requests.HTTPError:
        print("Error!")

download("https://www.cs220.com/vacation.html")
```
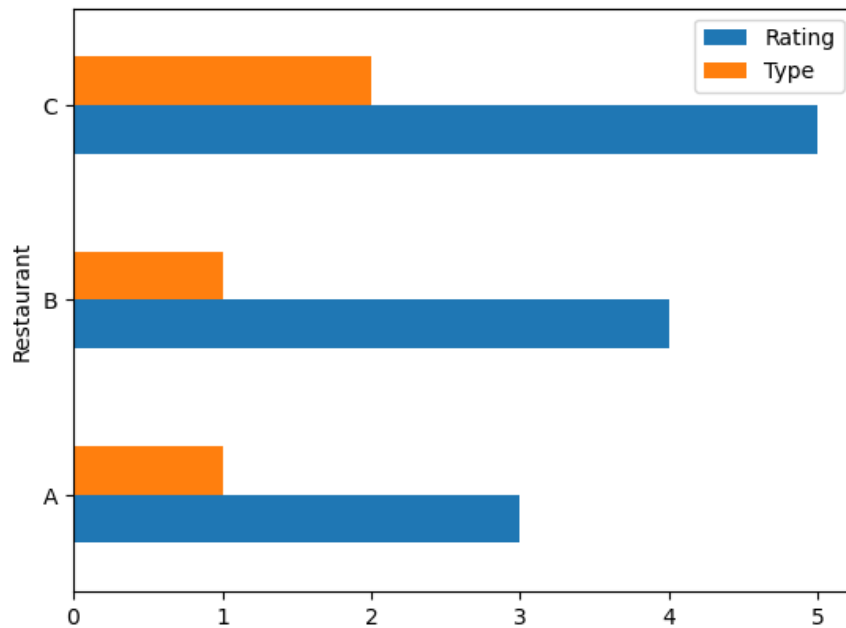
    A. `r.status_code`

    B. `r.raise_for_status()`

    C. `r.text`

    D. `r.read`

    E. `r.write`

27. Which of the following codes could generate this plot?

A. `ax = df.plot.barh()`
   `ax.set_ylabel("Restaurant")`

B. `df.plot.barh("Restaurant")`

C. `df = df.set_index("Restaurant")`
   `df.plot.barh()`

D. `All of the above could produce the plot depending on the`
   `construction of the DataFrame df`

28. Assume that all necessary imports have been made. Consider the following codes and the corresponding output:

```
movies_path = "movies.db"
base = sqlite3.connect(movies_path)

df = pd.read_sql("""
SELECT *
FROM movies
LIMIT 4
""", base)
df
```

–	Title	Genre	Director	Year	Runtime	Rating	Revenue
0	Guardians of the Galaxy	Action Adventure Sci-Fi	James Gunn	2014	121	8.1	333.13
1	Prometheus	Mystery Adventure Sci-Fi	Ridley Scott	2012	124	7.0	2012
2	Split	Horror Thriller	M. Night Shyamalan	2016	117	7.3	138.12
3	Suicide Squad	Action Adventure Fantasy	David Ayer	2016	123	6.2	325.02

Given that output, what is the output of the following code:

```
pd.read_sql("""
SELECT Title, Director, COUNT(*) as Count
FROM movies
GROUP BY Year
WHERE Revenue == 333.13
LIMIT 1
""", base)
```

A.

–	Title	Director	Count
0	Guardians of the Galaxy	James Gunn	1

B.

–	Title	Director	Count
1	Prometheus	Ridley Scott	1

C.

–	Title	Director	Count
3	Suicide Squad	David Ayer	1

D. The code with throw an error

Travel

For the next three questions, assume that the file "travel.html" exists in the current directory and contains the following content:

```
<h1>Space Travel</h1>
<ul>
    <li><a href="bali.html"> Bali </a></li>
    <li><a href="maldives.html"> Maldives </a></li>
    <li><a href="bora.html"> Bora Bora </a></li>
</ul>
```

29. Which of the following options when used to replace the ??? in the code below will retrieve the text saved in "travel.html"?

```
from bs4 import BeautifulSoup
import os

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

    A. `f.json()`

    B. `f.html()`

    C. `f.read()`

    D. `str(f)`

    E. `html(f)`

30. Assume that the code in the previous question ran successfully and the variables `html`, `doc`, and `link` are all defined. Then, what will be the output of the code snippet below?

```
print(link.get_text())
```

    A. `<a href="bali.html"> Bali </a>`

    B. `Bali`

    C. `["bali.html", "maldives.html", "bora.html"]`

    D. `bali.html`

31. Assume that the code in the previous two questions ran successfully and the variables `html`, `doc`, and `link` are all defined. Then, what will be the output of the code snippet below?

```
print(len(doc.find_all("li")))
```

    A. 0
    B. 1
    C. 2
    D. 3
    E. error

Stars and Planets

32. When handling file path strings, which of the following reasons best explains why it is good practice to use functions and objects from the `os` package - such as `os.path.join` and `os.path.sep` - instead of manually forming them and hardcoding separators like "/"?

    A. The `os` package has reduced runtime (in comparison to manually hardcoding) when forming path strings

    B. The `os` package will help generalize the file path strings to any operating system the code might be run on

    C. The `os` package has more flexible and powerful string manipulation functionality that is not present in base Python

    D. None of the above

33. Suppose in our current working directory, we have a directory 'data' with the following files inside:

stars_1.csv, stars_2.csv, planets_1.csv, planets_2.csv, mappings_1.json, mappings_2.json

Assume the files are those from Project10 and hold the corresponding stars/planets/mappings data. Consider the following code:

```
paths = [??? for file in os.listdir("data") if ??? and ???]
```

We want the variable paths to be a list of paths (relative to and accessible from the working directory) to only the csv files that hold the star data. Which of the following code snippets (in respective order) would best replace the ??? to accomplish this?

    A. `file, file.startswith("star"), (file is csv)`

    B. `file, file.startswith("star"), file.endswith(".csv")`

    C. `os.path.join("data", file), file.startswith("star"),`
       `file.endswith(".csv")`

    D. `"data/" + file, file.startswith("star"), file.endswith(".csv")`

    E. Both B and C

34. Consider the following code, and suppose we've defined the Star namedtuple as below, similar to Project 10:

```
from collections import namedtuple
star_attributes = ['spectral_type',
                   'stellar_effective_temperature',
                   'stellar_radius',
                   'stellar_mass',
                   'stellar_luminosity',
                   'stellar_surface_gravity',
                   'stellar_age']


Star = namedtuple("Star", star_attributes)

stars = ... # Assume stars is properly formed and populated by reading in data

filtered_star_ages = [??? for star in stars if ???]
print(sum(filtered_star_ages) / len(filtered_star_ages))
```

Assume we define a list of Star objects stored in the variable `stars`. We want the above code to print the average stellar age of all stars that have stellar mass of at most 5. You can assume that you will not encounter any missing values or Nonetypes. Which of the following code snippets could replace the ???, in respective order, to accomplish this?

    A. `star.stellar_age, star.stellar_mass <= 5`

    B. `star.stellar_mass <= 5, star.stellar_age`

    C. `stars[star].stellar_age, stars[star].stellar_mass <= 5`

    D. `star["stellar_age"], star["stellar_mass"] <= 5`

    E. Both A and D

35. What is the output of the following code?

```
from collections import namedtuple
star_attributes = ['spectral_type',
                    'stellar_effective_temperature',
                    'stellar_radius',
                    'stellar_mass',
                    'stellar_luminosity',
                    'stellar_surface_gravity',
                    'stellar_age']

Star = namedtuple("Star", star_attributes)

stars = [Star(0,0,1,0,0,3,0),
         Star(0,0,2,0,0,5,0),
         Star(0,0,1,0,0,1,0),
         Star(0,0,1,0,0,4,0),
         Star(0,0,4,0,0,7,0),
         Star(0,0,4,0,0,3,0)
        ]

m = min([s.stellar_radius for s in stars])
print(sum([s.stellar_surface_gravity for s in stars if s.stellar_radius == m]))
```

    A. 0

    B. 1

    C. 3

    D. 8

    E. 23

(Blank Page)