(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 (Gurmail)
   - 004 - MWF 9:55am (Mike)
   - 005 - MWF 3:30pm (Mike)
   - 006 - MWF 1:20pm (Gurmail)
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!

You may only reference your note sheet. You may not use books, your neighbors, calculators, or other electronic devices during 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 the exam and note sheet and your filled-in scantron form. The note sheet will not be returned.

General

1. Which of the following is mutable?

   A. tuple   B. string   C. dictionary   D. int

2. What is the output of the following code?

```
s = "Welcome to CS 220!"
s = s.split(" ")
print(s[2][-1])
```

   A. o   B. S   C. CS   D. 220!   E. IndexError

3. What will the following code print?

```
a = [1]
b = [2, 3]
c = [6, 7]
b = c
c.pop(-1)
b.append(a[-1])
print(a, b)
```

   A. [1] [2, 3]

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

   C. [1] [6, 1]

   D. [1] [6, 7, 1]

   E. [2, 3] [6, 1]

4. What is the output of the following code?

```
colors_dict = {"Red": 25, "Orange": 23, "Yellow": 32, "Brown": 7}
colors_dict[0] = "Green"
print(colors_dict[23])
```

   A. "Red"   B. 25   C. "Orange"   D. 23   E. KeyError

5. What is printed from the following code?

```
import copy
monsters_v1 = ["Frankenstein", ["Vampire", "Mummy", "Ghost", "Werewolf"]]
monsters_v2 = copy.deepcopy(monsters_v1)
monsters_v1[1][0] = "Dracula"
monsters_v1[0] = "Sand monster"
print(monsters_v2)
```

    A. ["Frankenstein", ["Vampire", "Mummy", "Ghost", "Werewolf"]]

    B. ["Frankenstein", ["Dracula", "Mummy", "Ghost", "Werewolf"]]

    C. ["Sand monster", ["Vampire", "Mummy", "Ghost", "Werewolf"]]

    D. ["Sand monster", ["Dracula", "Mummy", "Ghost", "Werewolf"]]

6. What is printed from the following code?

```
import copy
monsters_v1 = ["Frankenstein", ["Vampire", "Mummy", "Ghost", "Werewolf"]]
monsters_v2 = copy.copy(monsters_v1)
monsters_v1[1][0] = "Dracula"
monsters_v1[0] = "Sand monster"
print(monsters_v2)
```

    A. ["Frankenstein", ["Vampire", "Mummy", "Ghost", "Werewolf"]]

    B. ["Frankenstein", ["Dracula", "Mummy", "Ghost", "Werewolf"]]

    C. ["Sand monster", ["Vampire", "Mummy", "Ghost", "Werewolf"]]

    D. ["Sand monster", ["Dracula", "Mummy", "Ghost", "Werewolf"]]

7. What is the output of the following code?

```
total = 0
halloween_dict = {
    "Candy Corn": -1,
    "Caramel Apples": 2,
    "Jack O' Lanterns": 3,
    "Skeletons": 10
}

total += halloween_dict.pop("Candy Corn")
halloween_dict.pop("Caramel Apples")
total += halloween_dict.pop("Jack O' Lanterns")
halloween_dict["Ghosts"] = total
print(halloween_dict)
```

    A. {"Skeletons": 10, "Ghosts": 2}

    B. {"Skeletons": 10, "Ghosts": 4}

    C. {"Candy Corn": -1, "Caramel Apples": 2, "Jack O' Lanterns": 3, "Skeletons":10, "Ghosts": 2}

    D. {"Candy Corn": -1, "Caramel Apples": 2, "Jack O' Lanterns": 3, "Skeletons": 10, "Ghosts": 4}

8. What values are printed by the code below?

```
def f(n):
    if n > 5:
        f(n-1)
    print(n, end=" ")

f(9)
```

    A. 9 8 7 6

    B. 9 8 7 6 5

    C. 5 6 7 8

    D. 5 6 7 8 9

    E. The program runs into a `RecursionError`

9. What is the final value of `letter_list` and `number_list` after the following code is executed?

```
letter_list = ["B", "D", "E", "A", "C"]
letter_list = letter_list.sort(reverse = True)
number_list = [10, 50, 40, 20, 30]
number_list = sorted(number_list, reverse = True)
print(letter_list, number_list)
```

    A. `["E", "D", "C", "B", "A"] [50, 40, 30, 20, 10]`

    B. `["B", "D", "E", "A", "C"] None`

    C. `["E", "D", "C", "B", "A"] [10, 50, 40, 20, 30]`

    D. `None [50, 40, 30, 20, 10]`

    E. None None

10. Which of the following will produce a sorted dictionary based on the score of the players ordered from highest to lowest?

```
player_scores = {"Alice": 20, "Bob": 25, "Caroline": 21}
```

    A. `player_scores.sort(reverse = True)`

    B. `sorted(player_scores.items(), key = key[1], reverse = True)`

    C. `player_scores.sort(key = key[1])`

    D. `dict(sorted(player_scores.items(), key = lambda k:k[0], reverse = True))`

    E. `dict(sorted(player_scores.items(), key = lambda k:k[1], reverse = True))`

11. Which of the following would generate a **case-insensitive** list of words containing "s"?

```
lattes = ["Pumpkin Spice", "Coconut", "Pistachio", "Vanilla"]
```

    A. `[for latte in lattes:  if "s" in latte.lower():  latte]`

    B. `[latte if "s" in latte.lower() for latte in lattes]`

    C. `[latte for latte in lattes if "s" in latte]`

    D. `[latte for latte in lattes if "s" in latte.lower()]`

    E. `[latte for latte in lattes if "s" in latte.upper()]`

12. Consider the following code. What is the value of `list1` after the following code executes?

```python
list1 = []
list2 = [2, 1, 0, 4, 5]

i = 0
while i <= len(list2):
    try:
        c = 10 + list2[i]
        list1.append(c)
    except:
        list1.append(0)
    i += 1
```

   A. [0, 0, 0, 0, 0]

   B. [2, 1, 0, 4, 5]

   C. [12, 11, 10, 14, 15]

   D. [12, 11, 10, 14, 15, 0]

   E. The program runs into a `IndexError`

13. Assume the file `Hello.txt` does not exist before running this code. What is the value of the variable `data` after the following code executes?

```python
f = open("Hello.txt", "w")
f.write("Hello")
f.close()

f = open("Hello.txt", "w")
f.write("Hello")
f.write("World")
f.close()

f = open("Hello.txt", "r")
data = f.read()
f.close()
```

   A. "Hello"

   B. "HelloWorld"

   C. "Hello\nWorld"

   D. "HelloHelloHello"

   E. "Hello\nHello\nWorld"

Blank Page: this page is intentionally blank, please turn over for next section questions

Water Accessibility

For each question in this section, assume that the initial code executed is as follows:

```
pop_data = [
    {"country_name": "Albania", "country_code": "ALB",
        "pop_info": {
            2015: 2891,
            2016: 2900,
            2020: 2878
        }
    },
    {"country_name": "France", "country_code": "FRA",
        "pop_info": {
            2016: 64453,
            2020: 65274
        }
    },
    {"country_name": "Thailand", "country_code": "THA",
        "pop_info": {
            2015: 68715,
            2020: 69800
        }
    }
]
```

14. What does the following evaluate to?

    `pop_data[1]["pop_info"][0]`

    A. 2016   B. 2891   C. 64453   D. {2016 : 64453, 2020 : 65274}   E. KeyError

15. What does the following code evaluate to?

    `pop_data[-1]["country_code"][:-2]`

    A. ""   B. "F"   C. "T"   D. "HA"   E. "TH"

16. What will be printed when the following code is run?

```python
biggest_country = None
max_pop = None
year = 2016

for country in pop_data:
    if year not in country["pop_info"]:
        continue
    if biggest_country == None or country["pop_info"][year] > max_pop:
        biggest_country = country["country_name"]
        max_pop = country["pop_info"][year]

print(biggest_country)
```

A. Albania   B. France   C. Thailand   D. KeyError

17. What will get printed after we execute the below code? Read the code carefully. **Note**: The code has an intentional semantic error.

```python
buckets = {}                                    # Line 1
for country in pop_data:                        # Line 2
    for year in country["pop_info"]:            # Line 3
        year_pop = country["pop_info"][year]    # Line 4
        if year not in buckets:                 # Line 5
            buckets[year] = 0                   # Line 6
        else:                                   # Line 7
            buckets[year] += year_pop           # Line 8
print(buckets[2015])                            # Line 9
```

A. 0   B. 64453   C. 67353   D. 68715   E. 71606

18. Which of the lines from the code above should we change in order to correctly sum up the population for every year?

A. Line 4   B. Line 5   C. Line 6   D. Line 8

19. What will the following code print?

```python
pop_data.append(pop_data.pop(0))
print(pop_data[-1].get("country_name", None))
```

A. Albania   B. France   C. Thailand   D. None

Airbnb Venue

For the first three questions in this section, assume that the **initial code** executed is as follows:

```
from collections import namedtuple

Venue = namedtuple("Venue", ["name", "price", "neighborhood"])
listings = [
    Venue("The Ground Studio", 132, "Manhattan Beach"),
    Venue("Amazing Brownstone", 500, "Arden Heights"),
    Venue("Chelsea Living 2BR", 150, "Manhattan Beach"),
    Venue("Modern Garden Apartment", 107, "Long Island City")
]
```

20. What does `listings[-1].price` evaluate to?

    A. 107   B. "107"   C. 132   D. "132"   E. 150

21. Which line of code would change the "The Ground Studio" `neighborhood` to "Arden Heights"?

    A. `listings[0] = "Arden Heights"`

    B. `listings[0]["neighborhood"] = "Arden Heights"`

    C. `listings.neighborhood = "Arden Heights"`

    D. `listings[0].neighborhood = "Arden Heights"`

    E. `listings[0] = Venue("The Ground Studio", 132, "Arden Heights")`

22. What is the value of `names` after the following code executes?

    ```
    sorted_listings = sorted(listings, key = lambda t: -t.price)
    names = [v.name for v in sorted_listings]
    ```

    A. ['Modern Garden Apartment', 'Chelsea Living 2BR', 'The Ground Studio', 'Amazing Brownstone']

    B. ['Amazing Brownstone', 'Chelsea Living 2BR', 'Modern Garden Apartment', 'The Ground Studio']

    C. ['Modern Garden Apartment', 'The Ground Studio', 'Chelsea Living 2BR', 'Amazing Brownstone']

    D. ['Amazing Brownstone', 'Chelsea Living 2BR', 'The Ground Studio', 'Modern Garden Apartment']

23. Which of the following will create a dictionary mapping `name` to `price` for `Venue` objects with a `price` greater than or equal to 150?

    A. `dict([(v.name, v.price) for v in listings.items() if v.price >= 150])`

    B. `{v.name:v.price for v in listings if v.price >= 150}`

    C. `{v.name:v.price if v.price >= 150 for v in listings}`

    D. `[(v.name, v.price) for v in listings if v.price >= 150]`

24. Which of the following will **sort the original** `listings` **data structure** based on the length of the `Venue neighborhood`?

    A. `sorted(listings, key = lambda v:len(v.neighborhood))`

    B. `sorted(listings, key = def extract(v):len(v.neighborhood))`

    C. `listings.sort(key = lambda len(v:v.neighborhood))`

    D. `listings.sort(key = lambda v:len(v.neighborhood))`

Errors

25. What call to `foo` will **not** cause an `AssertionError`?

```
def foo(word1, word2):
    assert type(word1) == str and type(word2) == str
    assert len(word1) < len(word2)
    return (word1 + " is better than " + word2)
```

    A. `foo("penguin", "bear")`

    B. `foo(["seal"], "koala")`

    C. `foo("tiger", "elephant")`

    D. `foo("wolf", "frog")`

    E. `foo("zebra", ["pig"])`

26. What **line is printed last** when the following code is executed?

```python
my_list = [2, 0, 5, 0]

for item in my_list:
    try:
        print ("The number is:", item)
        amount = 1 / item
        print(amount)
    except ZeroDivisionError as e:
        print("help!")
```

A. The number is: 0   B. The number is: 5   C. `ZeroDivisionError`   D. help!

27. What is the **last line of code** that is executed in the following code?

```python
def fun1():                    # Line 1
    print("Start: fun1")       # Line 2
    best = "fun1" + 2          # Line 3
    print("End: fun1")         # Line 4

def fun2():                    # Line 5
    print("Start: fun2")       # Line 6
    try:                       # Line 7
        fun1()                 # Line 8
    except:                    # Line 9
        print("fun1 failed")   # Line 10
    print("End: fun2")         # Line 11

try:                           # Line 12
    fun2()                     # Line 13
except:                        # Line 14
    print("fun2 failed")       # Line 15
```

A. Line 10   B. Line 11   C. Line 14   D. Line 15

Files and I/O

28. What is the default delimiter between columns in a CSV file?

A. colon   B. comma   C. newline   D. semicolon

29. Which of the following is a valid JSON?

    A. {"Messi":{"Stats":'91, 93, 85', "Country":null, "Club":'PSG'}}

    B. {"Messi":{"Stats":"91, 93, 85", "Country":null, "Club":"PSG"}}

    C. {"Messi":{"Stats", [91, 93, 85], "Country", null, "Club", "PSG"}}

    D. {"Messi":{"Stats":[91, 93, 85], "Country":None, "Club":"PSG"}}

30. Suppose `double_check_exam.txt` has the following content:

```
Wisc ID number bubbled in
Special code matches exam
Answers filled in completely
```

What is the output of the below code snippet?

```
f = open("double_check_exam.txt")
for line in f:
    print(line[1])
f.close()
```

    A. Wisc ID

    B. Special code matches exam

    C. i
       p
       n

    D. ID
       code
       filled

    E. Wi
       Sp
       An

(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)