

CS 220 - Spring 2023

Instructors: Mike Doescher and Gurmail Singh

Exam 2 — 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 08:50 AM (Mike)
002 - MWF 11:00 AM (Mike)
003 - MWF 01:20 PM (Gurmail)
004 - MWF 03:30 PM (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 cannot 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. What is the value of `ans` after running the following code?

```
alpha = ["A", "B", "C", "D", "E", "F", "G", "H", "I"]
ans = alpha[1:5] [-2]
print(ans)
```

- A. ["A", "B"]
- B. ["A", "B", "C"]
- C. ["B", "C"]
- D. ["B", "C", "D"]
- E. None of the above

2. What is the data type of the variable `ans` from the previous question?

- A. int
- B. str
- C. list
- D. dict
- E. char

3. What is the value of `new_str` after running the following code?

```
str1 = "hello"
new_str = ""
count = 0
while count < len(str1):
    new_str += str1[count].upper()
    count += 2
```

- A. HLO
- B. hlo
- C. HELLO
- D. HL
- E. None of the above

4. Which of the following is mutable?

- A. string
- B. dictionary
- C. namedtuple
- D. list
- E. Both B and D

5. What is printed by the following code?

```
sentence = "Alice; Bob; Caroline; David"  
words = sentence.split("; ")  
print(words[3][-2])
```

- A. Caroline
- B. i
- C. ne;
- D. David
- E. None of the above

6. What is printed by the following code?

```
list1 = [2,5,1,-3]  
list1 = sorted(list1, reverse=True)  
list1.append(7)  
print(list1)
```

- A. [-3, 1, 2, 5, 7]
- B. [2, 5, 1, -3, 7]
- C. [5, 2, 1, -3, 7]
- D. [7, 5, 2, 1, -3]
- E. AttributeError: "NoneType" object has no attribute "append"

7. What is the output of the following code?

```
names = {"Alice": 1, "Bob": 2, "Caroline": 3}  
names["David"] = 4  
print(names[2])
```

- A. 2
- B. Bob
- C. Caroline
- D. {"Caroline" : 3}
- E. KeyError

8. What is the value of `alpha_list` after running the code below?

```
alpha_list = ["A", "B", "C", "D", "E"]
modified_list = alpha_list
modified_list[3] = "Z"
alpha_list[1] = "X"
```

- A. ["A", "X", "C", "D", "E"]
- B. ["A", "X", "C", "Z", "E"]
- C. ["X", "B", "Z", "D", "E"]
- D. ["A", "B", "C", "Z", "E"]
- E. `KeyError`

9. Select the answer that best fills in the two ... for the following statement:

In Python, CSV files are most commonly converted to the ... data structure and JSON files are most commonly converted to the ... data structure.

- A. `dictionary, dictionary`
- B. `list, list`
- C. `dictionary, list`
- D. `list, dictionary`
- E. `None of the above`

10. What is the value of `my_list` after running the following code?

```
my_dict= {"K":"v", "e":"a", "y":"l", "s":"s"}
my_list = []
for key in my_dict:
    print(key, end="")
    my_list.append(my_dict[key])
```

- A. []
- B. "Keys"
- C. ["v", "a", "l", "s"]
- D. ["K", "e", "y", "s"]
- E. "vals"

11. Which of the following creates a new reference to the `median` function object?

```
def median(items):
    sorted_list = sorted(items)
    if len(items) % 2 == 1:
        return sorted_list[len(items) // 2]
    else:
        first_middle = sorted_list[len(items) // 2]
        second_middle = sorted_list[(len(items) // 2) - 1]
        return (first_middle + second_middle) / 2

items = [1,4,2,3]
```

- A. `f = median(items)`
- B. `f = median()`
- C. `f = median([])`
- D. `f = median`
- E. None of the above

12. Given the following list of tuples `concerts` happening in the year 2023, what is the output of the below code snippet?

```
concerts = [("Taylor Swift", 6),
            ("Lil Wayne", 4),
            ("John Mayer", 3),
            ("P!nk", 8)]

concerts.sort(key=lambda item: item[1])
concerts[0][0]
```

- A. Taylor Swift
- B. Lil Wayne
- C. John Mayer
- D. P!nk
- E. None of the above

13. What is the output of the following code?

```
import copy

a = ["anchor", "boat", "sail", "rope"]
b = copy.copy(a)
y = copy.deepcopy(a)
a[1] = "train"
b[1] = "plane"
print(y[1])
```

- A. anchor B. boat C. plane D. train

14. Which of the following function calls will return a sorted list by the length of the string in ascending order?

```
def get_third(celeb):
    return celeb[2]
```

```
celebs = ["Taylor Swift", "Lil Wayne", "John Mayer", "P!nk"]
```

- A. sorted(celebs, key=len, reverse=True)
- B. sorted(celebs, key=len)
- C. sorted(celebs, key=get_third, reverse=True)
- D. sorted(celebs, key=get_third())
- E. Both B and D

15. Which of the following will return a dictionary of songs sorted by artist names alphabetically in descending order?

```
songs = {"Flowers": "Miley Cirus",
         "Lavender Haze": "Taylor Swift",
         "Players": "Coi Leray",
         "Emotionless": "Drake"}
```

- A. dict(sorted(songs.items(), key = lambda song:song[1], reverse = True))
- B. sorted(songs, reverse = True)
- C. dict(sorted(songs.items(), key = lambda song:song[1]))
- D. sorted(songs)
- E. Both A and C

Students

Consider the `students` data structure in the following questions.

```
students = [
    {
        "Name": "Alex",
        "Age": 19,
        "CGPA": 3.8,
        "Courses": "CS220, CS340, BS560",
        "Major": ["CS"]
    },
    {
        "Name": "Bob",
        "Age": 20,
        "CGPA": 3.6,
        "Courses": "CS220, CS630",
        "Major": ["CS", "Business"]
    },
    {
        "Name": "Emma",
        "Age": 21,
        "CGPA": 3.9,
        "Courses": "CS220, CS320, DS456",
        "Major": ["CS", "DS"]
    },
    {
        "Name": "Tina",
        "Age": 19,
        "CGPA": 3.7,
        "Courses": "CS220, CS540, CE569",
        "Major": ["Computer Engineering", "CS", "Design"]
    }
]
```

16. What is the value of `students[3]["Major"] [2]`?

- A. "Design"
- B. ["Computer Engineering", "CS", "Design"]
- C. "DS"
- D. "CS"
- E. IndexError

17. Which of the following would create a list of names of students who have more than one major?

- A. [student for student in students if len(student["Major"])>1]
- B. [student["Name"] if len(student["Major"])>1 for student in students]
- C. [student["Name"] for student in students if len(student["Major"])>1]
- D. [student["Name"] for student in students if student["Major"] > 1]

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

```
result=[]
for x in students[:2][1]:
    result.append(x)
print(result)
```

- A. ["N", "A", "C", "C", "M"]
- B. ["Name", "Age", "CPGA", "Courses", "Major"]
- C. ["Alex", "Bob", "Emma"]
- D. ["Bob", 20, 3.6, "CS220, CS630", ["CS", "Business"]]
- E. KeyError

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

```
age_bucket = {}
for item in students:
    bin_value = (int(item["Age"])//10) * 10
    if bin_value not in age_bucket:
        age_bucket[bin_value] = 0
    age_bucket[bin_value] += 1
print(len(age_bucket), age_bucket[20])
```

- A. 4 1
- B. 4 2
- C. 2 2
- D. 2 1

Books

For the questions in this section, consider the following data structure:

```
from collections import namedtuple

Book = namedtuple("Book", ["title", "author", "pages", "price"])
books = [
    Book("The Great Gatsby", "F. Scott Fitzgerald", 180, 9.99),
    Book("To Kill a Mockingbird", "Harper Lee", 240, 8.99),
    Book("1984", "George Orwell", 328, 7.99),
    Book("Pride and Prejudice", "Jane Austen", 352, 6.99)
]
```

20. Which of the following is the correct way to access the `pages` attribute of the first `Book` object in the `books` list?
- A. `books.pages[0]`
 - B. `books[0].pages`
 - C. `books[pages][0]`
 - D. `books[0]["pages"]`
 - E. `books[0][3]`
21. Which of the following lines will update the price of the "1984" book to 10.5 inside the `books` list?
- A. `books[2][3] = 10.5`
 - B. `books[2].price = 10.5`
 - C. `books[2] = Book(title = books[2].title, author = books[2].author, pages = books[2].pages, price = 10.5)`
 - D. `books[2]["price"] = 10.5`
 - E. None of the options, since the `books` list is immutable

22. What is the output of the following code?

```
new_books = sorted(books, key=lambda book: book.author.split()[-1])
print(new_books[0].title)
```

- A. "The Great Gatsby"
- B. "To Kill a Mockingbird"
- C. "1984"
- D. "Pride and Prejudice"

23. What is the output of the following code?

```
import copy
b1 = books[0]
b2 = copy.copy(books)
b2[0] = Book("The Catcher in the Rye", "J.D. Salinger", 224, 11.99)
print(b1.price)
```

- A. 9.99
- B. 11.99
- C. 6.99
- D. It will throw an error

24. What is the output of the following code?

```
import copy
b1 = copy.deepcopy(books)
b1.append(Book("Universe of Madness", "John Doe", 324, 13.99))
b1.sort(key=lambda b: b.title, reverse=True)
print(b1[0].title)
```

- A. The Great Gatsby
- B. To Kill a Mockingbird
- C. Universe of Madness
- D. 1984

25. What is the output of the following code?

```
def bar(x):
    if x <= 8:
        return False
    else:
        return True

def filter_books(books, scan):
    new_books = []
    for book in books:
        if not scan(book.price):
            new_books.append(book.title)
    return new_books

print(filter_books(books, bar))

A. ["The Great Gatsby", "To Kill a Mockingbird"]
B. ["The Great Gatsby", "Pride and Prejudice"]
C. ["1984", "Pride and Prejudice"]
D. ["1984", "To Kill a Mockingbird"]
```

Files & Errors

26. For which of the following would this code be valid syntax?

```
{  
    "Lecture1": {  
        "Enrollment": 0,  
        "Online": "true",  
        "Location": None,  
        "Students": []  
    },  
    "Lecture2": {  
        "Enrollment": 9,  
        "Online": "false",  
        "Location": "Bascom Hall",  
        "Students": [358, 965]  
    }  
}
```

- A. JSON File
- B. Python Dictionary
- C. A and B
- D. None of the Above

27. Consider the following function:

```
def checked_sum(lst):  
    sum = 0  
    for elem in lst:  
        assert type(elem) == int  
        sum += elem  
    assert sum > -10
```

Which of the following function calls will NOT cause an AssertionError?

- A. checked_sum([1, 2, [3, 4, 5]])
- B. checked_sum([-5, -6, -10])
- C. checked_sum([3, 4, 'a', 6])
- D. checked_sum([-6, 4, -7, 5])

28. Which line of code should be added to cause the “CS220” to print twice?

```
def h():
    ### ADD LINE OF CODE HERE ####
    print("CS220")
    try:
        x += 5
    except:
        print("CS220")

def g():
    try:
        h()
        x = 1 / 0
        print("CS220")
    except:
        print("CS220")

def f():
    try:
        print("CS220")
        g()
    except:
        print("CS220")

f()
```

- A. z = 0
- B. x = 0
- C. assert(False)
- D. print("CS220")
- E. None of the Above

29. Which lines of code are executed in the following code snippet?

```
numbers = [1,2,3,4]                                # Line 1
count = 0                                         # Line 2
try:
    for idx in range(len(numbers) + 1):           # Line 3
        count += numbers[idx]                      # Line 4
        print(numbers[idx])                        # Line 5
        assert count <= 0                           # Line 6
except IndexError:
    print("Accessing an invalid index in this list!") # Line 7
except AssertionError:
    print("The value inside count is incorrect!")   # Line 8
print("Exiting!")                                  # Line 9
```

- A. Lines 1, 2, 3, 7, 9
- B. Lines 1, 2, 3, 4, 7, 9
- C. Lines 1, 2, 3, 4, 5, 7, 9
- D. Lines 1, 2, 3, 4, 5, 6, 8, 9

30. How many times is Line 5 executed in the following code snippet?

```
numbers = [1,2,3,4]                                # Line 1
count = 0                                         # Line 2
try:
    for idx in range(len(numbers) + 1):           # Line 3
        count += numbers[idx]                      # Line 4
        print(numbers[idx])                        # Line 5
        assert count <= 0                           # Line 6
except IndexError:
    print("Accessing an invalid index in this list!") # Line 7
except AssertionError:
    print("The value inside count is incorrect!")   # Line 8
print("Exiting!")                                  # Line 9
```

- A. 0
- B. 3
- C. 4
- D. 5
- E. 8

(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)