

Sample Final Exam - **ANSWERS**

Note: this is taken from the final exam that I gave to the equivalent section in the Spring 2021 semester. This is slightly different because we covered slightly different material this semester, but the idea is the same.

Section 1: Multiple Choice/ True-False – 15 questions, 3 points each

1. Which of the following is not a valid Python variable name?

- a. `_time_of_day`
- b. `Time_of_day`
- c. `Time-of-day`
- d. All of the above are valid

ANSWER: c; hyphens are not allowed in Python variable names

2. The “big-O” of an algorithm such as a sorting algorithm indicates how the algorithm will perform in the average case

ANSWER: False; it tells how the algorithm will perform in the worst case

3. If a list is unsorted, binary search is always guaranteed to either find a searched-for item or determine that it is not present in $O(\log n)$ time.

ANSWER: False; this only applies if the list is SORTED

4. If you want to go through a list, item by item, and change certain values in the list (e.g., change every floating point number into an integer), which of the following loops could you use?

- a. for each loop
- b. for I loop
- c. while loop
- d. both for I loops and while loops would work

ANSWER: d. Using a for loop would cause errors to occur, but the other two loops could be used.

5. If you wanted to open a file for writing, but you did not want to delete any information that might be there, in what mode would you open the file?

- a. "a" or append mode
- b. "w" or write mode
- c. "r+" or reading and writing mode
- d. this cannot be done in Python

ANSWER: a - you would open the file in append mode

6. In CMSC 201, class coding standards require that you write a function so that it only returns one variable. What would be the BEST WAY to handle a case where you really, really wanted to return multiple values from a single function?

- a. Ignore the class coding standards; it's only a couple of points
- b. combine the values into a list, and return the list
- c. rewrite your code so you use multiple functions, each of which returns one of the values
- d. write your program without using functions at all and eliminate the problem

ANSWER: b - a list is considered a single value

7. Suppose I have the following definition:

```
def multiply_list(intlist, factor):  
  
    # this function multiplies each integer in intlist by factor and returns a list of the products  
  
    newlist = intlist  
  
    for i in range (len(newlist)):  
  
        newlist[i] = factor * newlist[i]  
  
    return newlist
```

this function is called with the statement

```
mylist = [1,2,3]  
  
fact = 4  
  
n = multiply_list(mylist, fact)  
  
print(mylist)
```

What is the result of the print statement

- a. An error; the statement fails to execute
- b. [1,2,3]
- c. [4,8, 12]
- d. [1, 4, 9]

ANSWER: the answer is c, but don't worry about this one. We didn't cover this material this semester

8. Now suppose I have the following definition:

```
def multiply_list(intlist, factor):  
  
    # this function multiplies each integer in intlist by factor and returns a list of the products  
  
    newlist = []  
  
    for i in range (len(newlist)):  
  
        newlist.append( factor * intlist[i])  
  
    return newlist
```

this function is called with the statement

```
mylist = [1,2,3]  
  
fact = 4  
  
n = multiply_list(mylist, fact)  
  
print(mylist)
```

What is the result of the print statement

- a. An error; the statement fails to execute
- b. [1,2,3]
- c. [4,8, 12]
- d. [1, 4, 9]

ANSWER: the answer is b, but don't worry about this one. We didn't cover this material this semester

9. True or False: you can create a dictionary by zipping together two lists of the same length.

ANSWER: True

10. What types can be used as values in a Python dictionary?

- a. only types like string, int and float
- b. only types like lists and dictionaries
- c. any type at all can be a value
- d. none of the other answers is correct.

ANSWER: c

11. Which of the following is a difference between lists and dictionaries in Python?

- a. lists are ordered; dictionaries are unordered
- b. Lists can be multi-dimensional; that is, be lists of lists; but dictionaries cannot contain dictionaries as values
- c. you can have a list whose elements are lists, but you can't have a dictionary whose values are dictionaries
- d. None of the above is correct

ANSWER: a is correct. B is false; you can have dictionaries as values. C is false; it's the same as b.

12. A file called numbers.py contains a python program I wrote last year. If I import numbers.py into my current python program, I can use code in the main program of numbers.py in my current program.

ANSWER: False. I can use code in the functions contained in numbers.py; but I can't use code from the main program.

13. A valid recursive program can have how many base cases?

- a. 0 or 1
- b. exactly 1

- c. at least 1
- d. any number

ANSWER: c. D is wrong because “any number” includes 0 and you can’t have 0 base cases.

14. Which is the correct way to instantiate a new, empty 2D list in Python?

- a. `l = [] []`
- b. `l = []`
- c. `l = { }`
- d. `; = { [] }`

ANSWER: b

15. If I import the Python math library into my program using the statement

`from math import *`

What is the correct way to call the arc tangent routine, `atan`?

- a. `Y = math.atan(x)`
- b. `Y = atan(x)`
- c. `Y = math(atan(x))`
- d. None of the other answers is correct

ANSWER: b.

Section 2: Short answer – 10 questions; 7 points each

16. Why is quicksort considered to be faster than bubble sort, even though in the worst case they’re both $O(n^2)$?

ANSWER: because $O(n^2)$ is the worst possible case for quicksort, and that’s extremely unlikely to happen. In the average case, quicksort will be about $O(n \log(n))$, which is much faster as n gets larger

17. When you are reading from a file, what does the file pointer do?

ANSWER: the file pointer indicates where the next value will be read from

18. Why is a sentinel while loop a good structure to use if you are getting input from a user at the keyboard? Think about whether the user will enter the value you want the first time he sees the prompt.

ANSWER: we have no idea how many times the user will run the loop. There may be any number of invalid values before a valid entry is made; and the user may want to run the loop any number of times. A sentinel works regardless of how many times the loop is to be run.

19. What can happen in a recursive program if the recursive case in a function does not involve a simpler version of the original problem?

ANSWER: an infinite loop or “too many recursions” can occur. If you don’t simplify the problem with each execution of the recursive case, you won’t ever get to a base case.

20. Suppose that s is a string with the value “Nebraska” Why is permissible to include a Python statement like

```
If s[0]== “N”:
```

But not a statement like

```
s[1] = “A”
```

ANSWER: strings are not lists; you cannot directly change a character in a string. You have to break the string into a list, change the list, and then join the list back into the new string.

21. Suppose you had a function, factorial, defined as follows:

```
def factorial (number):
```

```
    if number < 0:
```

```
        return -1
```

```
    elif number == 0:
```

```
        return 1
```

```
    else:
```

```
    return number * factorial (number -1)
```

And your main program was:

```
If __name__ == "__main__":
```

```
    n = 5
```

```
    k = factorial(n)
```

```
    print(k, "is the factorial of ", number)
```

Why would you get the error message that occurs?

ANSWER: “number” is a local variable in the function; it is not defined in the main program. It is out of scope when the main program is executing.

22. What characters are allowed to start a Python variable name?

ANSWER: upper and lower case letters, and the underscore

23. This semester we learned about a number of sorting algorithms including bubble sort and selection sort. The worst case performance of both is $O(n^2)$, but the best case performance of bubble sort is better – $\Omega(n)$ vs. $\Omega(n^2)$. Explain why.

ANSWER: Because you can stop bubble sort if the list is already sorted. If you go through the list and no swaps are made, you’re done. There’s no need to run through again. You can’t stop selection sort. If you go through the list and the current first element is the right element - no swap is made - you don’t know anything about the rest of the list. You still have to go through the list every time to make sure each element is in the right place.

24. Suppose that `states` is a one-dimensional list of state names in alphabetical order, and `senators` is a two-dimensional list of the 100 US senators. It has 50 rows; each row contains the names of the two Senators from that state. Write the Python code that lets you create a dictionary, `the_senate`, from these two lists. The keys are the state names; the values are the names of the two Senators from that state. You can presume that the two lists are in the proper order; e.g., the first row of `senate` contains the two Senators from the first state in `states`.

ANSWER:

```
senate_dict = dict(zip(states, senators))
```

25. Write a Python function that reads in a tab-separated value (.tsv) file as a string and splits it into a two-dimensional list. You may presume the name of the file is in the variable "file" and that .tsv files, like .csv files, have each line of data end in a newline character, "\n"

ANSWER:

```
def read_tsv_file(file):  
    with open (file, "r") as infile:  
  
        data = infile.read()  
  
        data_list = data.split("\n")  
  
        for i in range(len(data_list)):  
  
            data_list[i] = data_list[i].split("\t")  
  
    return data_list
```

Section 3: Programming – 3 questions, 15 points each

26. Write a Python program that asks the user to enter the user's semester grade in CMSC 201 as an integer between 0 and 1000, inclusive. If the user enters a number between 900 and 1000, inclusive, print out the message "congratulations on your A." If the user enters a number between 800 and 899, inclusive, print out the message "a B is not a bad grade." If the user enters a number above 1000 or below 800, print out the message "I don't believe that's a valid answer" and ask for the grade again. Continue doing this until you get a valid grade.

If the user enters something that is not an integer, it is okay if your program crashes.

.

27. Write a recursive Python program that reads a string from the last character to the first and prints out the message "found an x!" every time it encounters the letter "x" in the string. You MUST use a recursive function in your solution!

28. Insertion sort is an algorithm that sorts a list by first creating a new, empty list. Then it takes the elements of the unsorted list, one at a time, and inserts them in the proper place in the new list. That is: for each element in the original list, start at the beginning of the new list and find the first element that is greater than this new element, and insert the new element immediately before that. If there are no elements greater than this new element, append it to the end of the new list.

Write an interactive function in Python that implements insertion sort. You do not need to write the entire program; just the function. It should take one parameter, which is a list; and return a new, sorted list.