

CMSC 201 Final Review Sheet 2

1. Circle valid python variable names (some don't follow coding standards, but are valid)

`1Direction4Ever` `ILOVECMSC201` `_num_fru!ts` `DoGsRgOoD`
`thebestclassis201` `print` `Go0D1ucK0NtH#f|nAL!`

2. What gets printed? Does the following code snippet cause an error? Why or why not?

```
my_list = ["here", "are", "some",  
           "strings"]  
  
x = 4  
if x < len(my_list) and len(my_list) == 5:  
    print(my_list[x + 1])
```

No error, nothing gets printed, because of the if-statement

3. What gets printed on lines 8 and 10 and why?

```
1  def do_something(my_string):  
2      my_string = my_string.upper()  
3      my_list = my_string.split()  
4      my_string = "".join(my_list)  
5  
6  if __name__ == "__main__":  
7      my_string = "hello world"  
8      print(my_string)  
9      do_something(my_string)  
10     print(my_string)  
11  
12
```

"hello world", strings are immutable

4. Describe the difference between for and while loops.
For loops run for a specific amount of iterations, while loops don't
5. Why do we use Boolean flags?
readability
6. Why important to close the file after using it during file I/O?
To ensure other applications are allowed to use it
7. What is the difference between appending and writing to a file?
Appending appends to the end, writing writes to an empty file
8. What are the components of a dictionary? Describe their properties
Keys: like indexes, immutable, values are stored at a key
Values: like elements
9. What happens when a function is called?
Program executes function

10. Describe mutability, which data types are mutable/immutable, and applications of mutable data types.
If a data type is mutable, we can modify sections of it (dictionaries and list). We cant that with an immutable data type (ints, strings, floats, etc)
11. What is incremental development?
Developing little by little, testing as you go
12. Implement a recursive Fibonacci function
Pseudo code:
fib(n):
if n is 0 or 1
 return n
return fib of prev 2 numbers
13. Implement a pascal's triangle function.
Pseudo code:
Pascal(n)

For i in range(n)
for in range(i):
if outer element:
 print n on same line
otherwise
 sum up the prev. 2 elements from prior row and print it
14. Why would you use a dictionary over a list?
When using non-numerical indexes make more sense: ie proj 1
15. What is the output of this code snippet?

```
1     def count_vowels(word):
2         vowels = ["a", "e", "i", "o", "u"]
3         if word == "":
4             return 0
5         elif word[0] in vowels:
6             return count_vowels(word[1:]) + 1
7         else:
8             return count_vowels(word[1:])
9
10    if __name__ == "__main__":
11        word = "Elephants Are Great"
12        print("The Number of Vowels is ", count_vowels(word))
```

7

19. Given the following code, write the output

- a.

```
fact = "201 has the Coolest Professors, shhh!"  
print(fact[4:6] + fact[21] + fact[33:35])
```

harhh
- b.

```
fact = "201 students will do great on the exam if they try  
hard!"  
print(fact[4:8]+fact[46])
```

stud
- c.

```
fact = "Finally, the Important Things In Life That Matter!"  
print(fact[0:5]+fact[28]+fact[43:])
```

FinalsMatter!

20. What is the minimum number of base cases required for a recursive function? Minimum for recursive cases?

At least 1 base case, at least 1 recursive case

21. What is the correct order for the `range()` parameters?

- A. start, step, stop
- B. start, stop, step
- C. step, start, stop

22. Why can't you iterate through a dictionary with a loop? What can you use to iterate over a dictionary?

Dictionaries don't have a guaranteed order have to use a for each loop

23. What is the difference between sentinel values and boolean flags?

Sentinel values are constant, Boolean flags are not constant

24. Describe the best-case runtimes (and why) for the following: bubble sort, binary search, linear search, selection sort, insertion sort, quick sort

binary search- $O(1)$, item is found during 1st iteration

bubble sort- $O(n)$, list is sorted

linear search- $O(n)$, list is sorted

selection sort- $O(n^2)$, have to go through the list and find the smallest item for each sublist

insertion sort- $O(n)$, list is sorted

quick sort- $O(n\log_2 n)$, list is sorted, picked the a pivot in middle of list

25. In some situations, a recursive function will run until a `RecursionDepth` error occurs.

Why does this error occur, and *what* should be done to fix it?

Ran out of stack space, check base cases

26. List the fundamental differences between looping and recursion

Looping doesn't take up more space on the stack, recursion does

27. Explain the differences between `read()`, `readline()`, and `readlines()`. Give an example of when you might use each.

`read()` returns a single string of the file's contents

`readline()` returns the current string line of the file

`readlines()` returns a list of strings from the file

28. Recursively determine if a number is prime.

Pseudocode

`is_prime(n, m):`

If `n` is 0 or `n // m == 0`:

Return `true`:

if `n == 1` or `n==m`:

return `false`

increment `m`

return `is_prime(n, m)`

29. Recursively determine if a number is a power of `n`.

Pseudocode:

`is_power_of_n(n, num):`

If `num` is 1 or if `num = n`

Return `true`

If `num` is 0:

Return `false`

Otherwise

Return `is_power_of_n(n, num//n)`

30. List and explain the different file access modes.

`r`- read

`a`- append

`w`- write

31. Why do we care about runtimes?

Time and money

32. What's the difference between top-down and bottom-up development?

Top down → starting abstractly

Bottom up → devolving the details first

33. List string escape sequences.

`\t` → tab

`\n` → new line

`\\` → backslash

`\'` → single quote

`\"` → double quote

34. List and differentiate the different ways to access the keys of a dictionary.

`my_dictionary.get(key)` → returns value at key if key exists, returns None otherwise
`my_dictionary[key]` → returns value at key if key exists, throws error otherwise

35. Rank the following runtimes from **fastest** to **slowest**

- a. 1 , n , $\log_2 n$, $n \log_2 n$, n^2

For 36-38 you can use an online converter to find the answer

36. Convert the following decimal numbers to binary and hexadecimal:

- a. 463
- b. 63
- c. 31
- d. 255

37. Convert the following binary numbers to hexadecimal:

- a. 1010 0011 0101 1111
- b. 1101 1100 1011 0000
- c. 0000 0001 0010 0011
- d. 0110 1011 0101 1011

38. Convert the following hexadecimal numbers to binary and decimal:

- a. 14AD
- b. 002F
- c. 10BA
- d. FFFF
- e. 13EC