CMSC 201 Section 60
Fall 2023
Sample Final Exam

Notes:

1 This test is open book/open notes. You may NOT get help from any outside source; e.g., no talking to classmates and no communicating with people outside the classroom
2 You have two hours to complete this exam from when you are permitted to start
3 This test is worth ***170*** points

Section 1: True/False and Multiple Choice

There are 15 questions, worth 3 points each, for a total of 45 points. No partial credit will be given for problems in this section.

1. Which of the following is NOT a valid Python variable name?
    a. ThisIsASillyQuestion
    b. ***This-is-a-Silly-Question*** <- hyphens are not allowed
    c. This_is_a_silly_question
    d. None of the above is a valid Python variable name

2. Which of the following Python types are immutable and cannot be changed after being declared?
    a. Only dictionaries
    b. Dictionaries and lists
    c. ***Integers, booleans, floats and strings*** <- these are immutable
    d. None of the above is a correct answer

3. True or False: the Python symbol table does not contain the value of a variable; rather it points you to the address in memory where the value can be found.
    a. True
    b. False

4. True or False: the linear search algorithm only works on a list that is already sorted
    a. True
    b. False

5. What's the value of the expression
   ```
   False and 5 or not True
   ```
    a. True
    b. False
    c. 5
    d. It has no value; it's not a valid expression

6. What's the value of the expression
   ```
   (4//2+1) **2*6 - 3
   ```
    a. True
    b. False
    c. 51
    d. It has no value; it's not a valid expression

7. True or False: analysis of computer algorithms focuses on the worst-case behavior because with extremely large data sets, hitting that worst case can cause your entire program to fail to execute in a reasonable time.
    a. True
    b. False

8. The result of
   ```
   s = "abra ca dabra"
   s1 = s.split()
   s2 = "".join(s1)
   print(s2)
   ```
   Is "abracadabra"
   How would you change the code above to get what is printed out at the end to be identical to the original? *(Note that in each of the answers below, there is one blank space between the quotes.)*

    a. You can't; this is just a limitation of Python
    b. Make the second line read s1 = s.split(" ")
    c. Make the third line read s2 = " ".join(s1)
    d. Make the fourth line read print(s2, " ")

9. Which Python types can be valid keys in dictionaries?
    a. Lists and dictionaries
    b. Integers, booleans, floats and strings **<-** keys must be unique, and immutable
    c. Only integers and strings
    d. Any type at all can be a valid key

10. If l is a two-dimensional list, what does the 3 in `l[3][1]` refer to?
    a. The column number
    b. The row number
    c. The specific cell number
    d. None of the above

11. True or False: Every loop that can be written as a `for i l`oop can also be written as a `while` loop, but the reverse is not true
    a. True
    b. False

12. How many base cases are required for a valid recursive function in Python?
    a. At least one
    b. At most one
    c. Exactly one
    d. You don't need a base case, just a recursive case

13. Suppose that string `s = "abcdefghijklmno"`
    What is the result of `print(s[:5})  ?`
    a. "abcdefghijklmno"
    b. "5"
    c. "abcde"
    d. None of the above is correct

14. A global variable is::
    a. A variable defined within a Python function
    b. A variable defined in the main program
    c. Accessible from any part of a Python program
    d. b and c are both correct

15. Which of the following programming errors can cause a Python while loop to become an infinite loop?
    a. Forgetting to increment a variable within a loop body
    b. Setting a boolean condition in the while statement that can never be False
    c. Forgetting to initialize a variable using a priming read
    d. a. and b. are both correct.

Section 2: Short Answer

There are 16 questions, worth 5 points each, for a total of 80 points. All questions should be answered in no more than three sentences, or in a few lines of code. Partial credit WILL be awarded for answers that show an understanding of the material, even if those answers are not completely correct.

16.Think about the following Python program:

```
def multiply_by (l, n):
   for i in range(len(l)):
       l[i] *= n
   n *= n
   return

if __name__ == "__main__":
   nums = [1,2,3]
   factor = 4
   multiply_by (nums, factor)
   print (nums, factor)
```

When I run this, it produces

```
[4,8,12] 4
```

Yet there were no return values. Explain why, using the concept of mutable and immutable variables.

Since nums is a list, and lists are mutable values, any changes made to the parameter in the function are also made to the argument in the main program. Thus the for loop in the function changed the contents of nums in the main program.

17. Explain what “big O” means. That is, what does it mean to say that Big O of bubble sort is O(n^2)?

Big O represents the worst-case behavior of an algorithm; it means the algorithm will take "on the order of" that many operations. Bubble sort will take "on the order of n2" operations to sort a list

18. Integer variables in Python are immutable. They cannot be changed after being created. Given that fact, explain what happens in memory when Python executes the statements:

```
A = 4
A += 1
```

In the first statement Python allocates a new location in memory for A, stores the value 4 in that location, and enters the location information associated with A in the symbol table. When the second statement executes, Python fetches the current value of A, adds 1 to it to get 5, and then allocates a new location in memory to store that value. The symbol table is updated to store the new value .

19. What if anything is returned to the main program when a function ends with no return statement?

The special value None of type NoneType

20. Given a dictionary d, below:

```
d = {
   "Purdue":"B1G",
   "LSU":"SEC",
   "UMBC":"America East",
   "Navy":"Colonial",
   "Georgia":"SEC"
}
```

Is d a legal dictionary in Python? If not, why not?

Yes, it's legal

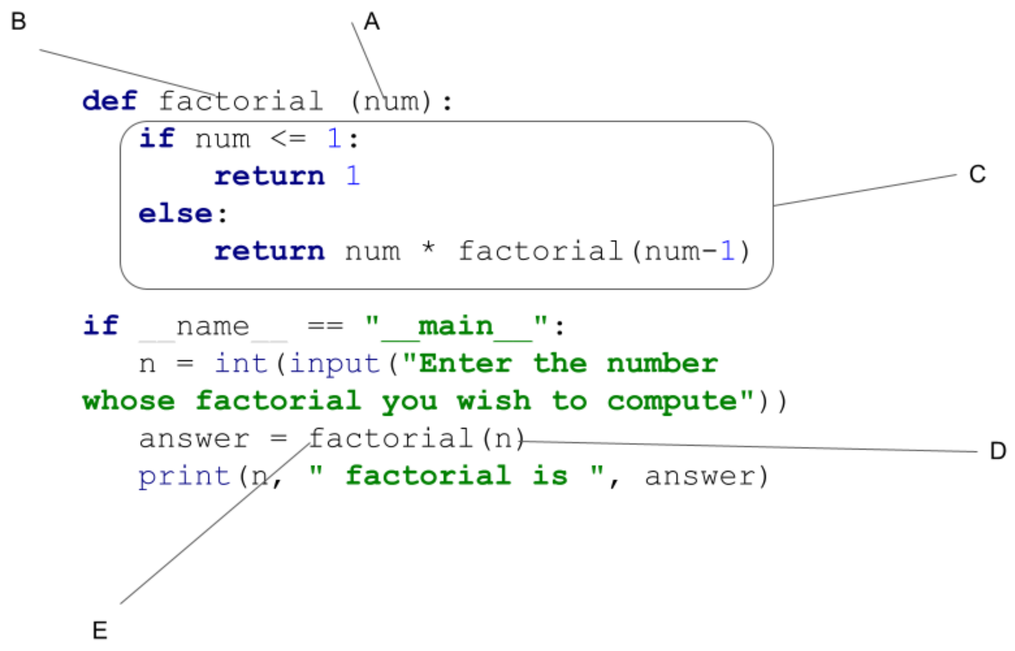
21. Explain what a sentinel while loop is.

A loop that executes until a particular value is encountered. That value is called the "sentinel."

22. What happens if you try to open up a file in read “r” mode in Python and the file does not exist?

You get an error message and the program crashes

23. Identify each labeled part of the program below:



```
def factorial (num):
    if num <= 1:
        return 1
    else:
        return num * factorial(num-1)

if __name__ == "__main__":
    n = int(input("Enter the number
whose factorial you wish to compute"))
    answer = factorial(n)
    print(n, " factorial is ", answer)
```

Function name/definition:____***B***__
Function body: __***C***___
Argument list _____***D***_______
Parameter list: ____***A***______
Function call: _____***E***_____

24. Trying to run the following Python code:

```
mylist = [1, 2, 3, 4]
mystring = “,”.join(mylist)
```

Produces an error message related to the join statement. What could you change to join the elements of the list?

You have to convert them to strings because join only works with lists of strings.

```
  for i in range(len(mylist)):
            mylist[i] = str(mylist[i])
```

Then execute the join statement

25. If l is a list, to what does l[-1] refer?

The last element in the list

26. Explain why 10/3 is not equal to 10//3.

10/3 is floating point division and yields 3.333333333… while 10//3 is integer division and yields 3.

27. Suppose I have a dictionary of my dogs' names and their breeds; namely:

```
dogs = {
   "doug":"beagle",
   "lizzie":"australian cattle dog",
   "baron": ["beagle","bulldog"]
}
```

How do I add another dog, "Penny" whose breed is "Spaniel" to this dictionary? You can just give the statement(s) if you prefer.

dogs[“Penny”] = “Spaniel”

28. Why is quicksort generally considered to be the fastest sorting algorithm we covered in class? (Compare to bubblesort or selection sort)

In the average case, quicksort runs in on the order of n * log(n) operations. That's much faster than n squared operations for large values of n.

For questions 29 and 30, use the following dictionary definition:

```
AL_east = {
   'Baltimore':["Orioles", "Camden Yards"],
   "New York": ["Yankees", "Yankee Stadium"],
   "Toronto": ["Blue Jays", "Rogers Center"],
   "Tampa Bay": ["Rays", "Tropicana Field"],
   "Boston": ["Red Sox"]

}
```

29. The entry for "Boston" does not include the name of their stadium, "Fenway Park." Write a line of Python code that would add "Fenway Park" to the "Boston" entry

AL_east["Boston"] = ["Red Sox", "Fenway Park"]

Or
AL_east["Boston"].append("Fenway Park")

30. The unthinkable has happened; the Tampa Bay Rays will move to Nashville and play in Music City Stadium. Write two lines of Python code that will delete the Tampa Bay entry in the dictionary, and add an entry for the Nashville Rays and Music City Stadium.

AL_east.pop("Tampa Bay")
AL_east["Nashville"] = ["Rays", "Music City Stadium"]

31. Suppose a is an integer variable in your program. Integers as we know are immutable variables in Python. Explain what happens in memory when you execute the following two statements in a Python program.

a = 4
a += 5

In the first statement Python allocates a new location in memory for a, stores the value 4 in that location, and enters the location information associated with a in the symbol table. When the second statement executes, Python fetches the current value of , adds 5 to it to get 9, and then allocates a new location in memory to store that value. The symbol table is updated to store the new value .

Section 3: Programming

There are three questions, worth 15 points each. Partial credit WILL be awarded for answers that show a significant understanding of the Python coding needed to solve the problems, even if those answers are not completely correct.

32. Write a Python program that prompts a user to enter her last name, a comma, and then her numeric semester grade - an integer - on one line. Then split the two values, add 5 points to the semester grade, and print out the student's name and revised semester grade. Your program should stop when the user types either a "q" or a "Q"

```
data = input("Enter your last name and numeric semester grade, separated by a comma")
while not data.upper() == "Q":
        data_list = data.split(",")
        data_list[1] = int(data_list[1])
        data_list[1] += 5
        print (data_list)
        data = input("Enter your last name and numeric semester grade, separated by a
comma")
```

33. Suppose I have a file called cities.txt that contains the names and populations of the five largest cities in Maryland, along with their population. There is one city/population pair per line; the name and population are separated by a tab. That is, the file looks like this:

```
Baltimore    576948
Columbia     105412
Germantown   89529
SilverSpring 81069
Frederick    79588
```

Write a Python program that reads in this file in the main program. Split it into a 2-dimensional list with each row having the city name (string) as an element and the population (integer) as an element. THEN: call a function that sums up the population figures and prints out the total. You MUST do this in a function. Write the entire program!t

```
def sum_population(cities):
        sum = 0
        for i in range(len(cities)):
                sum += cities[i][1]
        print (“The total population is”, sum)

if __name__ == “__main__”:
        with open (“cities.txt”, “r”) as infile:
                c = infile.readlines()
                for i in range(len(c)):
                        c[i].split(“\t”)
                        c[i][1] = int(c[i][1])
                sum_population(c)
```

34. Write a recursive Python function that compares two lists to see if one list is the other in reverse. Your function should take two parameters, each of which is a list; and return True (if one list is the other in reverse) or False (if not).

For example, if given the lists [1,2, 3] and [3,2, 1] your function should return True; if given [1,2,3] and [1,2,3] your function should return False.

You do not need to write the whole program; just the function. Think about base cases where the answer is obviously True, or where the answer is obviously False. Then think about recursive cases and how to simplify the decision.

```
def check_reverse(list1, list2):
        if len(list1) != len(list2):
                return False
        if len(list1) == 0:
                return True
        If list1[0] != list2[-1]:
                Return False
        else:
                return (check_reverse(list1[1:], list2[:-1])
```