CSC2-140 Foundations of Computer Science
Exam 2 Part 1 October 9, 2019

Solution

name

This exam has 2 parts and this is part 1. This part is worth 60 points. Closed book, closed notes. You may not use any devices while taking this part of the exam.

1. (2 pts. each) Fill in the blanks.

a. A data type whose values cannot be changed (modifying functions create a totally new object that does not change the original one) is called a(n) Immutable data type.

b. A function which has no side effects rather it only make changes to the calling program through their return values is called a(n) pure function.

c. A(n) slice is a part of a string (substring) specified by a range of indices.

d. You must open a file before you can read its contents. When you are done with a file, you should close it.

e. In computer programming, aliasing refers to the situation where the same memory location can be accessed using different names.

f. The first computer programmer, who programmed Charles Babbage's Analytical Engine in the 19th Century and whose "Day" we celebrated yesterday is Ada Lovelace

g. Encapsulation describes the idea of bundling data and methods that work on that data within one unit. This concept is also often used to hide the internal representation, or state, of an object from the outside.

2. (7 pts.) What is the value of each of the following expressions:

a. 'Python'[1] y

b. "Strings are sequences of characters."[7] —

c. len("Python") 6

d. 'Mystery'[:4] Myst

e. 'p' in 'Pineapple' True

f. 'pear' not in 'Pineapple' True

g. 'purple' < 'Peach' False

3. (6 pts.) Write a code segment that will allow a user to enter a sentence and will print out the number of characters and the number of words in that sentence. (For example, if the user entered "Good morning.", your program would print "Your sentence had 13 characters and 2 words."

```
sent = input("Enter your sentence.")
print("your sentence had ", len(sent), " characters
    and ", len(sent.split()), " words.")
```

for loop OK too

4. (5 pts.) Write a code segment that will create a tuple called odds containing all the odd integers between 1 and 999 inclusive. *Hint: recall that, for example, a tuple holding the integer 7 has to be represented as (7,) rather than just (7).*

```
odds = tuple(range(1, 1000, 2))
```

for loop OK too

5. (2 pts. each) What is printed by the following code segments?

a.
```
ger = {'one':'eins', 'two':'zwei', 'three':'drei'}
del ger['one']
print(len(ger))
```
2

b.
```
ger = {'one':'eins', 'two':'zwei', 'three':'drei'}
print('two' in ger)
```
True

c.
```
ger = {'one':'eins', 'two':'zwei', 'three':'drei'}
print(ger['five'])
```
Error

6. (8 pts.) Consider the following 2 code segments, similar in structure:

```
#segment 1
s = 3
t = s
s = s + 1
print (t)
```

3


```
#segment 2
this = ["do", "re", "me", "fa"]
that = this
this[2] = "sol"
print (that)
```

Produce a reference diagram for each code segment. What will be printed by this code? Explain the difference.

7. (5 pts.) Consider the following function. For each function invocation, what is printed? Briefly explain.

```
def change ( myList):
    myList.append('The End')
```

a.
```
demoa = [1,2,3]
change(demoa)
print (demoa)
```

[1, 2, 3, "The End"]

b.
```
demob = ["a","b","c"]
newList = change(demob)
print (newList)
```

None

8. (3 pts.) Write a Python statement that would create a dictionary to translate texting acronyms to their full string. Include these three—BTW is "by the way", TTYL is "talk to you later", and OMG is "oh my goodness".

AC = {"BTW": "bytheway", "TTYL": "talk to you later", "OMG": "oh my goodness"}

9. (4 pts.) Give what is printed out by the following code segments

a.

```
aList= ["Hello","Goodbye"]
other = aList[:]
print(aList == other)
print (aList is other)
```

True
False

b.

```
str = 'Hello'
print(str.find('h'))
print(str.find('o'))
```

-1
4