

COSC 101, Exam #2

March 2019

Name: _____

Please write your name above. Do not start the exam until instructed to do so.

You have 50 minutes to complete this exam.

There are 5 questions and a total of 47 points available for this exam. Don't spend too much time on any one question.

Since indentation is important in Python, please be sure that your use of indentation is obvious for any code you write.

If you want partial credit, show as much of your work and thought process as possible.

If you run out of space for answering a question, you can continue your answer on one of the blank pages at the end of the exam. If you do so, be sure to indicate this in two places: (1) below the question, indicate which blank page contains your answer, and (2) on the blank page, indicate which question you are answering.

Question	Points	Score
1	7	
2	6	
3	5	
4	9	
5	20	
Total:	47	

1. (7 points) Assume that the following statements have already been executed:

```
a = 'March 14'  
b = 5  
c = 'Apple Pie'
```

For each of the following expressions, evaluate the expression and write the resulting value, or identify the error in the code that would prevent it from running.

- (a) `c[-3]`

Solution:

`'p'`

- (b) `a[2] + 'are'`

Solution:

`'rare'`

- (c) `c[:3]`

Solution:

`'App'`

- (d) `int(a[6:12] + b)`

Solution:

Error: must be str, not int

- (e) `len(a)`

Solution:

8

- (f) `a[::2]`

Solution:

`'Mrh1'`

- (g) `a + " is " + c[6:8].upper() + " Day"`

Solution:`'March 14 is PI Day'`

2. (6 points) What is the output of the following program?

```
def doSomething(word, scale):  
    result = ''  
    for i in range(len(word)):  
        for j in range(scale,0,-1):  
            result += word[:i+1] * j + "\n"  
    return result  
  
print(doSomething("to", 3))
```

Solution:

```
ttt  
tt  
t  
tototo  
toto  
to
```

3. (5 points) Write a function that given two strings `str1` and `str2`, and an integer `ind`, will return a new string with the character at index `ind` in `str1` replaced with the value of `str2`.

For example, if `str1` is "sos", `str2` is "ea" and `ind` is 1, the function should return the string "seas".

Do not use the string method `replace`. For full credit the body of your function should not use any loops.

Solution:

```
def indexReplace(str1, str2, ind):  
    if ind >= 0:  
        return str1[:ind] + str2 + str1[ind+1:]  
    else:  
        ind = ind + len(str1)  
        return str1[:ind] + str2 + str1[ind+1:]
```

4. Consider the program below:

```
1 def mystery(d):
2     prompt = "----- "
3     x = int(input(prompt + str(d) + ": "))
4     while x % d != 0:
5         print("Input Invalid. Try again.")
6         x = int(input(prompt + str(d) + ": "))
7     return x
8
9 if mystery(4) % 4 == 0:
10     print("4")
11 elif mystery(3) % 3 == 0:
12     print("3")
13 else:
14     print("neither")
```

(a) (2 points) Briefly, what is the purpose of the mystery function?

Solution:

(b) (2 points) What would you write instead of the blank line for the variable prompt (on line 2) to describe what the function expects from the user?

Solution:

(c) (1 point) For what value of d would this function not work properly?

Solution:

(d) (4 points) What is displayed on the console when the program is run if the user enters 9 and then 12 at the prompts?

Solution:

```
1 ----- 4: 9
2 Input Invalid. Try again.
3 ----- 4: 12
4 4
```

5. A popular website wants to try out a new game with its users in order to get users to read their content more closely. The game involves one of the sentences in the three most recent news posts being scrambled. You will write some functions to aid them in implementing their new game.

- (a) (5 points) Write a function `scramble` that will take a word and scramble its letters. You may assume that a single word with no punctuation is passed to the function. Your function should be written such that all possible combinations of the letters in the word are possible outcomes. As a refresher, the `randint(a, b)` function from the `random` module will return an integer between `a` and `b`, inclusive.

For example,

```
print(scramble('hello'))
```

might output

```
lelho
```

or any other combination/ordering of the letters in `hello`.

Solution:

```
import random

def scramble(word):
    scrambled = ""
    while len(word) > 0:
        ind = random.randint(0, len(word)-1)
        scrambled += word[ind]
        word = word[0:ind] + word[ind+1:]
    return scrambled
```

- (b) (7 points) Write a function that given a sentence (a string) and an integer `num` can return word number `num` in the sentence, where word numbers start at 1. You may assume that the sentence input has had all punctuation removed. You may not use and string methods.

For example,

```
print(get_word_number_x('Today is a great day!', 4))
```

would output

```
great
```

The first three lines of the function are

```
def get_word_number_x(sentence, num):  
    word_count = 0  
    last_space = -1
```

For the remainder of the function select one line of code from each of the pairs of lines of code given and reorder them to solve the problem:

```
A1         if word_count == num:  
A2         if word_count < num:  
B1     return sentence[last_space+1:]  
B2     return sentence[last_space:i]  
C1         if sentence[ch] == ' ':  
C2         if sentence[i] == ' ':  
D1                 return sentence[last_space+1:i]  
D2                 return sentence[last_space:i+1]  
E1         word_count += 1  
E2         word_count = 1  
F1     for i in range(len(sentence)):  
F2     for ch in sentence:  
G1         last_space = 0  
G2         last_space = i
```

Select only seven lines of code from above, and only one line from each pair. You may fill in line identifiers (*e.g.*, E2) on the next page, or write out the code.

Solution:

```
for i in range(len(sentence)): # F1
    if sentence[i] == ' ': # C2
        word_count += 1 # E1
        if word_count == num: # A1
            return sentence[last_space+1:i] # D1
        last_space = i # G2
return sentence[last_space+1:] # B2
```

- (c) (8 points) Write a function that takes a sentence (a string) as a parameter, and using the two previous functions returns that sentence with its words scrambled. You may assume the two previous functions work correctly as described (regardless whether or not your particular solution is correct). You may further assume that the sentence the website staff selects has punctuation only at the end of the sentence.

Solution:

```
def scramble_sentence(sentence):  
  
    scrambled_sentence = ""  
    next_word_num = 1  
  
    while len(sentence) - 1 > len(scrambled_sentence):  
        next_word = get_word_number_x(sentence[:-1], next_word_num)  
        scrambled_sentence += scramble(next_word) + " "  
        next_word_num += 1  
  
    return scrambled_sentence[:-1] + sentence[-1]
```

(This page is intentionally blank. Label any work with the corresponding problem number.)

(This page is intentionally blank. Label any work with the corresponding problem number.)