

CMPS 8 W18
Introduction to Computer Science

Midterm 1 Examination

Please state your answers as clearly as possible. This exam not only tests your understanding of the material, but also how well you can convey your understanding to us. Remember that you are solely responsible for the answers to the questions, therefore, please refrain from consulting with your class peers.

Please write all your answers **LEGIBLY** and **CLEARLY**. If we cannot decipher your answers, you will not receive credit.

No electronic devices are allowed during the exam (calculators, cell phones, laptops, etc.).

READ all questions carefully before attempting to answer. If there are any ambiguities in the statement of questions, please ask us. **You may assume that each problem is correct and solvable unless the question specifically asks about errors.**

THE GRADE IN THIS EXAM IS A TOTAL OF 68 POINTS.

Name (as it would appear on the official course roster)	Umail Address
	@umail.ucsb.edu

Question 1 (10 points)

Use the following variable definitions to answer this question:

```
count = 10
s = "kitten"
animals = ["raccoon", "dog", "cat", "goldfish", "snake", "lizard"]
```

Provide the **EXACT** output as it executes the following statements. Write your answer in the box to the right of the statements.

<code>print(s, count)</code>	
<code>print(s * 2)</code>	
<code>print(count + len(animals))</code>	
<code>print(count <= (len(animals) + 5))</code>	
<code>print(animals[4])</code>	
<code>print(animals[-2])</code>	
<code>print("gold" in animals)</code>	
<code>print("ten" in s)</code>	
<code>print(len(s) / 2)</code>	
<code>print(animals[1:3])</code>	

Question 2 (4 points)

Use the following variable definitions to answer this question:

```
class1 = "CMPSC8"  
class2 = "CMPSC16"  
school1 = "UCSB"  
school2 = "UCLA"
```

Provide the **EXACT** output displayed on the Python interactive shell when typing the following statements. Write your answer in the space to the right of the statements.

<code>class1 < class2</code>	
<code>"8" in class1 or "8" in class2</code>	
<code>len(class1)==6 and len(class2)==4</code>	
<code>not class2[-2] in school1</code>	

Question 3 (6 points)

Check all of the legal Python variable names on the line to the left of the names below:

<input type="checkbox"/> <code>'x'</code>	<input type="checkbox"/> <code>_x</code>
<input type="checkbox"/> <code>variable</code>	<input type="checkbox"/> <code>a-name</code>
<input type="checkbox"/> <code>2places</code>	<input type="checkbox"/> <code>VarX</code>

Question 4 (4 points)

What does the following code print out? Write the output in the blank box to the right of the code.

```
s = "friends"  
count = 10  
print("Looking for", s)  
if count < 10:  
    print("Not a lot of friends")  
    count = count - 8  
else:  
    print("Lots of friends!")  
    count = count - 5  
print(count, s[count:count+2])
```

Question 5 (6 points)

- Briefly state (in at most two sentences) what “syntax” means.
- Briefly state (in at most two sentences) what “semantics” means.
- Briefly state (in at most two sentences) what a python module is.

Question 6 (10 points)

Using the following variable assignments below:

```
s = "ant"
animals = ["raccoon", "dog", "cat", "goldfish", "snake", "lizard"]
ints = [8, 6, 7, 2]
```

Identify the **data type** of each of the following expressions, using definitions that have appeared in previous questions. Write the type next to each statement. Choose from **int, float, bool, str, list of str, list of int**

len(animals[0:2])	_____	ints[2]	_____
"Jane Doe"	_____	animals != ints	_____
animals[-1]	_____	ints[0] / ints[-1]	_____
animals	_____	animals * ints[-1]	_____
ints.pop()	_____	s > animals[2]	_____

Question 7 (6 points)

Using the following function definition below:

```
def f(x):
    return 4 / 2 + 2 ** x - 1
```

Check all of the `assert` statements that will **fail** on the line to the left of the statements below:

_____ <code>assert f(2) == 4</code>	_____ <code>assert f(4) == 17</code>
_____ <code>assert f(2) == 5</code>	_____ <code>assert f(4) == 0</code>
_____ <code>assert f(2) == 0</code>	_____ <code>assert f(4) == 10</code>

Question 8 (8 points)

Complete the following function definition such that the assertions below will pass:

```
def hasStringLength(n, stringList):
    ''' The parameter n is an integer and stringList is a list of strings. Return
        True if stringList contains a string with length n. Return False
        otherwise '''
```

```
assert hasStringLength(2, ["a","b","abc"]) == False
assert hasStringLength(3, ["a","b","abc"]) == True
assert hasStringLength(1, ["a","b","abc"]) == True
```

Question 9 (8 points)

What does the following code print out? Write the output in the blank box to the right of the code. Be **exact** in your answer for full credit.

```
for x in range(4, 16, 3):
    if x % 2 == 1:
        print(x)
    else:
        print(x / 2)
```

Question 10 (6 points)

Given the following python code below:

```
from collections import namedtuple

# Item namedtuple represents a grocery item where name is the item's name, price
# is the cost of a single item, and purchased is the quantity purchased.

Item = namedtuple('Item', 'name price purchased')

def computeRevenue(item):
    ''' Returns the revenue made (quantity * price) of an item '''
    return item.purchased * item.price

i1 = Item('Eggs', 1.50, 2)
i2 = Item('Milk', 4.00, 1)
i3 = Item('Honey', 3.00, 3)
i4 = Item('Sugar', 2.50, 4)
i5 = Item('Orange', 0.50, 2)
i6 = Item('Avocado', 2.00, 1)
```

Fill in the blanks with the correct values to make the following `assert` statements pass. Choose from `i1`, `i2`, `i3`, `i4`, `i5`, `i6`.

```
assert computeRevenue(_____) == 10      assert computeRevenue(_____) == 4
assert computeRevenue(_____) == 3      assert computeRevenue(_____) == 9
assert computeRevenue(_____) == 1      assert computeRevenue(_____) == 2
```