COSC 101, Exam #1
26 September 2017

Name: ______________________________ Section: 8:30 / 9:55 / 1:20 / 2:45

Instructions and Advice:

- Do not open the exam until instructed to do so.
- Write your name and circle your section.
- You have 60 minutes to complete this exam, use your time wisely.
- There are 5 questions and a total of 50 points available for this exam. Don't spend too much time on any one question.
- If you want partial credit, show as much of your work and thought process as possible.
- Since indentation is important in Python, please be sure that your use of indentation is obvious for any code you write.
- It is not necessary to write comments.
- If you run out of space while answering a question, you can continue your answer on one of the scrap 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 scrap page contains your answer, and (2) on the scrap page, indicate which question you are answering.

Question	Points	Score
1	8	
2	12	
3	8	
4	10	
5	12	
Total:	50	

1. (8 points) Assume that the following statements have already been executed:

```
a = 3
b = False
c = """12"""
d = 4.7
e = "apple"
```

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 + " " +e + "s or " + str(d) + " " + c + "s"`

Solution:

```
'12 apples or 4.7 12s'
```

(b) `a / c`

Solution:

```
Error: unsupported operand type(s) for /: 'int' and 'str'
```

(c) `d + 2 * a > int(c) == b`

Solution:

```
False
```

(d) `c + 3`

Solution:

```
Error: cannot concatenate 'str' and 'int' objects
```

(e) `not(e + c == c + e)`

Solution:

```
True
```

(f) `str(int(d) // 2) + str(int(d) % 2)`

Solution:

```
'20'
```

(g) `float(c) + a + d`

Solution:

```
19.7
```

(h) `b and not(d > a)`

Solution:

```
False
```

2. (a) (4 points) What is the output of the following program?

```
pumpkins = 240
people = 105
minperperson = 1

if (pumpkins < people * minperperson):
    print("We need more pumpkins!")
elif (pumpkins >= people * minperperson):
    print("We have enough pumpkins.")
    pumpkinsperperson = pumpkins // people
    if (pumpkinsperperson >= 1):
        print("Each person may carve", pumpkinsperperson, "pumpkin(s).")
    else:
        print("At least", 2 * people - pumpkins, "people must share a pumpkin.")
    if (pumpkinsperperson == 1 and pumpkins % people == 0):
        print("There will be no extra pumpkins.")
    else:
        print("There may be extra pumpkins.")
else:
    print("We have bigger problems than pumpkins!")
```

Solution:

```
We have enough pumpkins.
Each person may carve 2 pumpkin(s).
There may be extra pumpkins.
```

(b) (4 points) What is the output of the following program?

```
for i in range(4):
    print("i:", i+1)

for j in range(4,0,-1):
    print("j: "+str(j))
```

Solution:

```
i: 1
i: 2
i: 3
i: 4
j: 4
j: 3
j: 2
j: 1
```

(c) (4 points) What is the output of the following program, if the user enters `4` and `happy`?

```
cnt = int(input("Enter a whole number: "))
instr = input("Enter a string: ")

mystr = "Be"

for i in range(cnt):
    mystr = mystr + instr * i + "!" * (cnt-i)
    print(mystr)
```

Solution:

```
Be!!!!
Be!!!!Happy!!!
Be!!!!Happy!!!HappyHappy!!
Be!!!!Happy!!!HappyHappy!!HappyHappyHappy!
```

3. (8 points) Write a program that asks two users for their names, ages, and heights. The your program will print the name of the user who is older (e.g. `"Joe is older than Alice."`) and the name of the user who is shorter (e.g. `"Alice is shorter than Joe."`). If the there is a tie print that (e.g. `"Joe is the same age as Alice."`). Assume that heights are entered in inches and all ages are entered as whole numbers. Your program should match these examples:

```
USER 1:                           USER 1:
- Name: Josue                     - Name: Cal
- Age: 19                         - Age: 20
- Height: 70                      - Height: 67

USER 2:                           USER 2:
- Name: Sneha                     - Name: Pei-Yi
- Age: 21                         - Age: 22
- Height: 68                      - Height: 67

Sneha is older than Josue.        Pei-Yi is older than Cal.
Sneha is shorter than Josue.      Cal is the same height as Pei-Yi.
```

Solution:

```
print("USER 1:")
name1 = input("– Name: ")
age1 = int(input("– Age: "))
height1 = float(input("– Height: "))

print()

print("USER 2:")
name2 = input("– Name: ")
age2 = int(input("– Age: "))
height2 = float(input("– Height: "))

print()

if age1 > age2:
    print(name1, "is older than", name2+".")
elif age2 > age1:
    print(name2, "is older than", name1+".")
else:
    print(name1, "is the same age as", name2+".")

if height1 > height2:
    print(name2, "is shorter than", name1+".")
elif height2 > height1:
    print(name1, "is shorter than", name2+".")
else:
    print(name1, "is the same height as", name2+".")
```

4. (10 points) Suppose you are training for a race and you decide to write a Python program to help you train. Your program should ask the user how long the race is (in miles), how long until the race, and how many miles they are currently running per week. Then your program will output how many miles they should be running each week until the race, with up to two decimal places. Your mileage should increase the same amount each week until you are running three times the race distance in the final week.

   Your output should exactly match the format of this example:

```
How long is the race? 26.2
How many weeks until the race? 5
How many miles did you run last week? 10

This week you should run 23.72 miles.
In week 2 you should run 37.44 miles.
In week 3 you should run 51.16 miles.
In week 4 you should run 64.88 miles.
In week 5 you should run 78.6 miles.
```

Solution:

```
race_length = float(input("How long is the race? "))
weeks_away = int(input("How many weeks until the race? "))
miles_run = float(input("How many mniles did you run last week? "))

final_mileage = race_length * 3
increase_mileage = round((final_mileage - miles_run) / weeks_away, 2)
miles_to_run = miles_run

print()
for week in range(1, weeks_away+1):

    miles_to_run += increase_mileage

    if week == 1:
        print("This week you should run", miles_to_run, "miles.")
    else:
        print("In week", week ,"you should run", miles_to_run, "miles.")
```

5. (12 points) You are managing a grocery store and must keep apples in stock but don't want to count by hand. Write a program that estimates how many apples are left at the end of the day and prints a message telling the user whether they need to order more apples to have at least 50 in stock for the next day.

   Your program should first ask the user how many apples were in stock at the beginning of the day. Then your program should ask how many customers bought apples that day and how much each customer's order of apples weighed. Your program needs to estimate how many apples each customer bought, assuming the average weight of one apple is 0.33 pounds and partial apples are not for sale. (Hint: The math module has a function called ceil that rounds a number up to the nearest integer.)

   Your output must exactly match these examples:

```
How many apples were there? 80
How many customers? 2

Customer 1 apple weight: 20
Customer 2 apple weight: 5.5

There are about 2 apples left.
Order more apples.
```

```
How many apples were there? 100
How many customers? 3

Customer 1 apple weight: 4.2
Customer 2 apple weight: 6.1
Customer 3 apple weight: 2.3

There are about 61 apples left.
No need to order more yet.
```

Solution:

```python
import math

# initialize variables
avewght = 0.33 #pounds
maxapples = 100
apples_purchased = 0

# get the number of apples at the start of the day from the user
instock = int(input("How many apples were there? "))

# get the number of purchases from user
customers = int(input("How many customers? "))
print()

# ask user about each customer
for i in range(customers):
    w = float(input("Customer " + str(i+1) + " apple weight: "))
    apples_purchased += math.ceil(w/avewght)

# calculate how many apples are left
```

```
new_stock = instock - apples_purchased
print("\nThere are about", int(new_stock), "apples left.")

# print whether to order more apples
if (new_stock < 50):
    print("Order more apples.")
else:
    print("No need to order more apples yet.")
```

(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.)

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