SI@UCF Introduction to Programming in Python Test #1 Solutions
6/16/2017

1) (6 pts) Write a single statement in python the produces the following output:

```
Qian's son said,
"Hello!"
```

```
print("Qian's daughter said,\n\"Hello!\"")
```

Grading: 1 pt print, 1 pt reg text, 2 pts escape newline, 2 pts escape ' or ".

2) (10 pts) What are the values of the following expressions in Python?

a) `13 - 2*4` **5**

b) `12345%100` **45**

c) `1 + 2 - (3*4//5)` **1**

d) `16 - 6*(12/(3 + 65%12)` **7**

e) `27 + 17000//18902 + 570000%19` **27**

Grading: 2 pts, all or nothing, for each part

3) (12 pts) Complete the program below so that it reads in the length, width and height of a rectangular box(prism), in inches, and prints out the surface area of the box (all 6 sides) of the box, in square inches.

```
def main():

    length = int(input("What is the length, in inches, of your box?\n"))
    width = int(input("What is the width, in inches, of your box?\n"))
    height = int(input("What is the height, in inches, of your box?\n"))

    sArea = 2*(length*width+length*height+width*height)

    print("The box's surface area is", sArea , "sq. in.")

main()
```

Grading: 3 pts for valid syntax for the implementation of any expression, 6 pts total if they output the volume, 8 pts total if they have the correct dimensionality (sq in) but the wrong formula, 10 pts if they just forgot to multiply LW+LH+WH by 2, take off a single point for really simple errors

4) (12 pts) UCF gives aid to students whose annual income is less than $30,000 and UCF also gives aid to students taking more than 17 credit hours. No other students get aid. (Note: This scenario is completely made up.) Complete the program below so that it asks the user(student) how much their annual income is and how many credit hours they are taking, and prints out whether or not the student receives aid.

```
def main():

    income = int(input("What is your annual income?\n"))
    hrs = int(input("How many credit hours are you taking?\n"))

    if income < 30000 or hrs > 17:
        print("You get aid.")
    else:
        print("Sorry, you don't get aid.")

main()
```

Grading: 2 pts for if, 1 pt for income < 30000, 2 pts for or, 1 pt for hrs > 17, 2 pts aid print, 2 pts else, 2 pts sorry print. Take off 2 pts if the prints are switched.

5) (15 pts) In a single fruit fly season, the number of flies doubles (due to births) followed by 20 deaths. For example, if the season started with 80 fruit flies, they would double to 160, but then 20 would die, leaving 140 at the end of the season. Complete the program below so that given the initial number of fruit flies and the number of seasons they are cultivated, it ***prints out the total number of fruit flies alive at the end of each of those seasons.*** (For example, if we started with 80 fruit flies and cultivated them for two years, there would be 260 fruit flies at the end of the second season, since 2 x 140 - 20 = 260, so your program should print out 140, then 260.)

```
def main():

    initFlies = int(input("How many flies are there initially?\n"))
    numYears = int(input("How many years are they cultivated?\n"))

    for season in range(1,numYears+1):
        numFlies *= 2
        numFlies -= 20
        print("Season: ",season,", Flies: ",numFlies,sep="")

main()
```

6) (10 pts) What is the output of the following Python program?

```
def main():
    a = 13
    b = 4
    for i in range(5):
        c = 4*b + a
        print(a,b,sep=",",end=" : ")
        b = c//3
        a = a - c%a
main()
```

```
13,4 : 10,9 : 4,15 : 4,21 : 4,29 :
```

Grading: 1 pt per value, no exceptions

7) (20 pts) Write a python program below using the turtle so that it asks the user to enter a positive integer n and draws n squares, each of side length 25, right next to each other, in a horizontal row. The leftmost square should have its lower left pixel value be (0, 0). You must use a loop to get full credit. For a little extra credit, fill every ***other*** square (the 2nd, 4th, 6th, etc.) with the color red.

```
import turtle

def main():

    n = int(input("How many squares do you want?\n"))

    for i in range(n):

        for j in range(4):
            turtle.forward(25)
            turtle.left(90)

        turtle.forward(25)

main()
```

Note: Extra Credit is posted online in the file test1_q7ec.py

Grading: no main necessary, 3 pts read input, 3 pts loop running n times (or whatever variable they created), 12 pts for drawing a square (2 pts right size, 3 pts above x axis, 3 pts correct turn angle, 4 pts rest - no loop necessary for the square), 2 pts for moving forward to the new starting position.

For the Extra Credit, award up to 5 extra points if they did it correctly. Add partial as you deem fit.

8) (10 pts) Recall that the function random.randint(a,b) returns a random integer in between a and b, inclusive. Complete the program below so that it prints out a strictly increasing list of 200 random integers. To ensure the list is strictly increasing, generate the next number in the sequence by taking the previous one and adding a random integer in between 1 and 99, inclusive, to it. Let the very first number in the list just be a random integer in between 1 and 99, inclusive. For a bit of extra credit, after you write your program, determine the expected value of the last number printed.

```
import random

def main():

    last = 0

    for i in range(200):
        cur = last + random.randint(1, 99)
        print(cur)
        last = cur

main()
```

Extra Credit - Expected Value of Last Number Printed = **10000**

Grading: 2 pts for setting last (or equivalent), 2 pts for the loop, 2 pts for correct random call in loop, 2 pts for adding it to last and printing that, 2 pts for updating last

Extra Credit: Add 2 points if they get it, all or nothing.

9) (5 pts) What tasty breakfast item is Donut King's specialty? **Donuts** (Give to all)