SI@UCF Introduction to Programming in Python Test #1 Solutions
6/14/2019

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

```
One\
"Two"
'Three'
```

print("One\\\n\"Two\"\n'Three'")

Grading: 1 pt print, 2 pt reg chars, 1 pt \\, 1 pt \n, 1 pt \"
Note: we can flip the role of ' and ".

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

a) `14 + 2*3` **20**

b) `917%200` **117**

c) `(3 + 2*7)//5` **3**

d) `6 + 10/8 - (48%49)/(4**3)` **6.5**

e) `13 + 22220//2223 + 1500%500` **22**

Grading: 2 pts each all or nothing

3) (12 pts) David has figured out a new scheme to make money. He buys a few dozen (a dozen is 12) donuts from Dunkin Donuts and then sells them to his classmates for $1.50 each and collects a handsome profit. Unfortunately, some crafty students have found a way to get a donut from David without paying. Complete the program so it reads in how many dozen donuts David bought, the price for one dozen donuts, and how many people didn't pay him for their donut, and computes David's profit. (You may assume that he still makes a profit.)

```
DOZEN = 12
SELLPRICE = 1.50
numDozens = int(input("How many dozen donuts did David buy?\n"))
priceDozen = float(input("What is the cost of a dozen donuts?\n"))
numNoPay = int(input("How many students didn't pay for their donut?\n "))

numPay = DOZEN*numDozens - numNoPay
profit = numPay*SELLPRICE - priceDozen*numDozens

print("David made $",profit, "selling donuts.", sep="")
```

Grading: 4 pts first line or equivalent, 6 pts second line, 2 pts correct variable

4) (12 pts) Each set of 10 jumping jacks burns 7 calories and each set of lunges burns 4 calories. Complete the program below so that it reads how many calories the user wants to burn, the number of sets of lunges they've done and calculates and outputs the minimum number of sets of 10 jumping jacks he has to do to meet his calorie burning goal.

```
JUMP_CAL = 7
LUNGE_CAL = 4
totalCalories = int(input("How many calories do you want to burn?\n"))
numLunges = int(input("How many lunges have you done?\n"))

totalCalories -= LUNGE_CAL*numLunges

print("You need to do",(totalCalories+JUMP_CAL-1)//JUMP_CAL,"sets jj")
```

Grading: 5 pts for subtracting out calories already burned,
4 pts for division of calories left by JUMP_CAL
3 pts for edge cases (ceiling function)

5) (15 pts) Complete the program below so that it prints out the numbers from start (entered by the user) to end (also entered by the user) except that if a number is divisible by 5, print the word "Fizz", if a number is divisible by 7, print the word "Buzz", and if it's divisible by both, print "FizzBuzz". Print one item (a number, "Fizz", "Buzz" or "FizzBuzz") per line. A sample input and output will be given on the board.

```
start = int(input("What is the starting number?\n"))
end = int(input("What is the ending number?\n"))

for i in range(start, end+1):

    flag = False
    if i%5 == 0:
        print("Fizz",end="")
        flag = True
    if i%7 == 0:
        print("Buzz", end="")
        flag = True
    if not flag:
        print(i,end="")
    print()
```

Grading: 3 pts for loop,
3 pts for printing Fizz if divisible by 5
3 pts for printing Buzz if divisible by 7
3 pts for printing number if not divisible by either
3 pts for rest (new lines, order, etc.)

Note: We can check 35 as a separate case to solve the problem also.

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

```
a = 1
b = 3
for i in range(10):
    c = b + a
    print(a,end=" ")
    a = b
    b = c
```

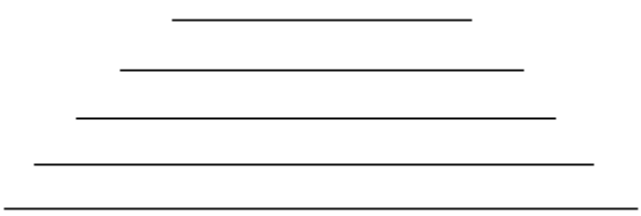
1 3 4 7 11 18 29 47 76 123

7) (12 pts) Recall that the function random.randint(a,b) returns a random integer in between a and b, inclusive. Write a program that simulates rolling a pair of fair six-sided dice 500 times, keeps track of the total number of doubles rolled and outputs this number. (Note: doubles are when both dice show the same number.)

```
import random


TRIALS = 500
count = 0

for i in range(TRIALS):

    die1 = random.randint(1, 6)
    die2 = random.randint(1, 6)
    if die1 == die2:
        count += 1

print("You rolled",count,"number of doubles.")
```

Grading: 1 pt setting up accumulator
2 pts loop
4 pts generating both dice
2 pts if
2 pts update accumulator in if
1 pt print at end

8) (20 pts) Write a python program below using the turtle so that it asks the user to enter a positive integer n, another positive integer gap, a third positive integer start, and a fourth positive integer skip and prints out a design of n parallel horizontal lines, such that the top line has length start, and the line below the previous line is drawn gap pixels below and has a length that is 2*skip longer than the previous line. A sample design with n = 5, gap = 20, start = 100 and skip = 15 is included below. (Note: this is just an approximation.) In the drawing, the top line has length 100 pixels, the line right below it has length 130 pixels and starts at an x value 15 less than the top left x value and starts at a y value 20 less than the top line's y value.



You can choose which pixel value to start at, but everything in the drawing has to be properly relative to that.

```
import turtle
n = int(input("How many lines to draw?\n"))
gap = int(input("How many pixels between the lines?\n"))
start = int(input("What is the length in pixels of the top line?\n"))
skip = int(input("How much does each subsequent line extend left and right
compared to the previous line in pixels?\n"))

x = -100
linelen = start + 2*skip*(n-1)
for y in range(-100, -100+n*gap, gap):

    turtle.penup()
    turtle.setpos(x, y)
    turtle.setheading(0)
    turtle.pendown()
    turtle.forward(linelen)

    x +=skip
    linelen -= 2*skip
```

```
Grading: Looping the correct # of times - 4 pts (3 pts off by 1)
         Dealing with penup, pendown correctly - 4 pts
         Changing correct variable each time in loop (y) - 2 pts
         Correct spacing for y - 2 pts
         Correct line length - 4 pts (2 or 3 pts if inconsistency but a
                               Similar pattern)
         Correct heading - 2 pts
         Correct changes for x - 2 pts
```

9) (3 pts) Food from what country is served at Café de France? **French**