

COP 2930 Exam #1 Day #1 Solution
2/19/2020

1) (5 pts) Write a single print statement to produce the following output:

```
Happy
Birthday\\
George
```

print("Happy\\nBirthday\\\\\\\\\\nGeorge")

Grading: 1 pt print, 1 pt (), 1 pt "", 1 pt reg stuff, 1 pt escape

2) (5 pts) What is the value of the following arithmetic expressions?

a) $3 + 8 * 2$ **19**

b) $7 // 4 + 3$ **4**

c) $7 / 4 + 3$ **4.75**

d) $13 \% (2 + 6)$ **5**

e) $1000 \% 1001 + 1000 // 1001$ **1000**

Grading: 1 pt for each, must be exactly correct to get the pt.

3) (5 pts) Which of the following is not a valid name for a variable? Why? Note: There may be more than one that is not valid. To get full credit, you must identify each invalid variable name and why each is not valid. (Circle the invalid ones and give the reasons on the lines provided.)

a) variable

b) 4get

c) score5

d) forget

e) continue

4get is not a valid variable name because it starts with a digit, which isn't allowed, continue isn't allowed because it's a reserved word in Python.

Grading: 1 pt b, 1 pt reason b, 1 pt e, 1 pt reason e, 1 pt nothing else.

4) (8 pts) It's important to make code easy to read for others because in most work environments, many people may have to edit the same code base. Explain four techniques one can use to make their code more readable. Please explain each technique clearly.

Here is a partial list of acceptable ideas to list for this problem:

1. Header Comment – Let's the reader know the main point of the program, who wrote it and when.

2. Internal Comments – English that explains what each section of code does makes it easier for another reader to make sense of code.

3. Meaningful Variable Names – The variable name “cost” is better than “x”, since the name itself indicates the function of what it's storing.

4. Good use of White Space – Blank lines between logical sections of code indicate to the reader the separation between sections of code which do different things and also allow the brain to digest code in sizable chunks.

5. Try not to combine too many steps in a single line of code, so that the reader can easily follow the steps.

6. Don't change the loop index inside of the loop.

Grading: 2 pts for each reason, partial credit possible for each reason.

5) (6 pts) What is the output of the following segment of code?

```
x = 5
y = 4
area = x*y
print(x,y,area)
x = x + 2
y = y + 1
print(x,y,area)
```

5,4,20

7,5,20

Note: While both x and y get updated between the two prints, area isn't, so it stays the same as it was previously.

Grading: 1 pt for each value, must get it exactly to get the problem.

6) (4 pts) What is the largest integer the following segment of code could print out?

```
x = int(input("Enter an integer. "));

if x > 30:
    print(50)
elif x < 20:
    print(40)
elif x < 70:
    print(60)
elif x > 0:
    print(100)
else:
    print(200)
```

Answer: 60

For all $x > 30$, 50 gets printed. So, now, let's only consider values ≤ 30 . For all $x < 20$, 40 gets printed. This accounts for all numbers except 20, 21, ..., 29, 30. All 11 of these numbers are less than 70, so 60 gets printed for these cases, but the last two cases never trigger.

Grading: 4 pts for 60, 2 pts for 50, 0 pts for any other answer.

7) (4 pts) How many times does "Go Knights!" print when the code segment below is executed?

```
for i in range(17,100,10):
    print("Go Knights!");
```

Answer: 9

i starts at 17, then skips by 10, so i assumes the values 17, 27, ..., 87, and 97. There are 9 numbers on this list. (We can see this as there is one number starting with each digit 1 through 9.)

Grading: 4 pts for 9, 3 pts for 8 or 10, 2 pts for 7 or 11, 0 pts otherwise.

8) (4 pts) How many times does "Go Knights!" print when the code segment below is executed?

```
for i in range(100):
    print("Go Knights!");
    if i//5 > 2:
        break
```

Answer: 16

When $i = 15$, it's the first time $i//5$ is greater than 2. Thus, AFTER printing, when $i=15$, the loop terminates. Thus, "Go Knights" prints for $i=0, 1, 2, \dots, 15$. This is 16 times.

Grading: 4 pts for 16, 3 pts for 14, 15 or 17, 2 pts for [10,13], [18,20], 0 pt otherwise.

9) (4 pts) How many times does “Go Knights!” print when the code segment below is executed?

```
for i in range(100):  
    if i%10 > 2:  
        continue  
    print("Go Knights!");
```

Answer: 30

When $i\%10$ is 3,4,5,6,7,8 or 9, the continue statement is run and no print occurs. This means that when $i\%10$ is 0, 1 or 2, the print occurs. Since we go through ten sets of 10: [0, 9], [10, 19], ..., [90, 99], for each set of ten, the message prints out 3 times, so the total times it prints is $10 \times 3 = 30$, since there are 10 sets of 10.

Grading: 4 pts for 30, 3 pts for [20, 29] [31, 40], 2 pts for [10, 19], [41, 50], 0 pts otherwise

10) (4 pts) What does the following code segment print out? (Note: more lines than necessary are provided.)

```
for i in range(1,5):  
    for j in range(i):  
        print(i, end="")  
    print()
```

1
22
333
4444

i goes through the values 1 through 4. Since we print i, we'll always print the same number inside the j loop. Since there is a print after the j loop, all the numbers on a line will be the same. Since the inner loop runs i times, it follows that 1 will be printed once, 2 will be printed twice, and so forth.

Grading: 1 pt for having 4 rows of anything, 1 pt for there being 1 item on row 1, 2 items on row 2, 3 items on row 3 and 4 items on row 4, 2 pts for getting all the values correct, 1 of these pts if it says 1, 12, 123, 1234.

11) (1 pt) By what acronym is the Ladies Professional Golf Association known? LPGA

Grading: Give to all.