

## ***Comp 150 Exam 2 Overview.***

### **Resources During the Exam**

The exam will be closed book, no calculators or computers. You may bring notes on two sides of 8.5x11 inch paper (either both sides of one sheet, or two sheets written on single sides). Write this as you study! I mostly want to test you on concepts, not memorized rote facts.

### **Main topics that may be on exam 2: Required sections of the Python Tutorials through while loops, section 3.3.4**

1. Control flow: sequential, decision if-elif-else, loop through sequence, while, functions calls with parameters and return statements. Only *read* while loops, not write them for this exam.
2. Creating a new list, append; the len function for sequences.
3. The range function with 1, 2 or 3 parameters.
4. String methods lower and upper from the beginning of Chapter 2
5. Print variants, with the keyword parameters sep and end.
6. Converting types between int and string.
7. Files – Input: opening, read all; Output: open, write, close
8. Simple nested loops.
9. graphics: methods for GraphWin and graphics objects: using getMouse and creating, drawing, cloning and moving Points, Circles, Lines, Rectangles, and Polygons. (No other methods.)
10. Boolean values, expressions with comparisons and with operations 'and', 'or', 'not'; using the Boolean result.

### **How the Python topics get used:**

1. Follow fairly arbitrary code using the elements above, and show the results. Distinguish exactly what is the output from the sequence of internal steps.
2. Write a few lines of code translating ideas into Python; put several steps together.

### **Read the following before looking at either the problems or the solutions!**

**(Both points are the same as from the Exam 1 Review)**

1. Study first, gathering your written notes. Look at the chapter summaries and start by filling in any holes. Then look at the sample problems. The sample problems cannot give complete coverage, and if you look at them first, you are likely to study just these points, and will not get an idea how well you are prepared in general.
2. Do not look at the answers until you have fully studied and tried the problems and gotten *help* getting over rough spots in the problems if you need it! Looking at the answers before this time makes the problems be just a few more displayed examples, rather than an opportunity to actively learn by doing and check out where you are. The *doing* is likely to help you be able to *do* again on a test.

*New sample problems start on the next page.*

## Review Problems for Exam 2 (Solutions follow the problems.)

1. Suppose the file 'prob1.txt' contains the two lines

Hello

Mom

What is printed by

```
fin = open('prob1.txt', 'r')
```

```
s = fin.read()
```

```
print(s.upper())
```

2. What will be the contents of the file prob2.txt? Indicate any blanks or newlines clearly.

```
fout = open('prob2.txt', 'w')
```

```
words = ['Hello', 'there', 'Mom']
```

```
for w in words:
```

```
    fout.write(w)
```

```
fout.close()
```

3. What will be printed by the function calls in parts a-d?

```
def comp(x):
```

```
    if x < 3:          #1
```

```
        print("A")    #2
```

```
    elif x > 10:       #3
```

```
        print("B")    #4
```

```
    else:
```

```
        print("C")    #5
```

a. comp(5) b. comp(12) c. comp(-2) d. comp(10)

4. What will be printed by the function calls in parts a-d?

```
def comp2(x, y):
```

```
    if x == y:          #1
```

```
        print("A", end='') #2 empty end each time
```

```
    elif x < 5 and y > 2: #3
```

```
        print("B", end='') #4
```

```
    if x > 2 or y > 4:    #5
```

```
        print("C", end='') #6
```

a. comp2(5, 3) b. comp2(5, 5) c. comp2(1, 5) d. comp2(1, 1)

5. What is printed? Here **end** is one space.

```
x = 1          #1
```

```
while x < 5:    #2
```

```
    print(x, end=' ') #3
```

```
    x = x + 2     #4
```

6. What is printed? Carefully follow the execution sequence! Here **end** is one space.

```
for x in [30, 40]: #1
```

```
    for y in [1, 2, 3]: #2
```

```
        print(x+y, end=' ') #3
```

```
    print()             #4
```

9. Write code that inputs a number from the user and prints "High" if it is over 100, "Low" if it is less than 50, and "In between" otherwise.

7. What is printed? Here **end** is one space.

```
for n in [1, 3]:          #1
```

```
    for s in ['a', 'b']:  #2
```

```
        print(s*n, end=' ') #3
```

8. What is printed by the Python code?

```
nums = list()             #1
```

```
for i in range(4):        #2
```

```
    nums.append(2*i)      #3
```

```
print(nums)               #4
```

10. Assume you have a GraphWin called win. Write code to draw a circle of radius 10 and center at the point (40, 50)

11. Complete the function definition.

```
def double(numlist):  
    '''One number to a line, print twice each number in the numlist.  
        For example double([3, 7, 4]) prints  
        6  
        14  
        8  
        '''
```

12. Modify the previous problem so it prints out a sentence stating the multiplication fact for each number. .  
For example the example above would print

```
Twice 3 is 6.  
Twice 7 is 14.  
Twice 4 is 8.
```

Use a format string.

13. Complete the Python function below.

```
def printWords(wordlist):  
    '''Print on one line the words in wordlist.  
    For example, if words is ['he', 'is', 'his', 'hero'],  
    printWords(words) prints: he is his hero    '''
```

14. Suppose num, lowVal, and highVal are variables with existing numeric values, and lowVal <= highVal.  
Write an expression that is True if num is in the interval from lowVal to highVal, allowing the endpoint  
values lowVal and highVal. For instance, if lowVal is 2 and highVal is 5, your expression should be True if  
num is 2, 3, 4.4 or 5, but false if num is -1, 1.9, 5.1, 7 or 100000.

15. Complete the function definition.

```
def numbersBetween(numList, lowVal, highVal):  
    '''Print on one line the numbers in numList that lie in the  
        interval from lowVal to highVal, allowing lowVal and highVal  
    For example,  
    numbersBetween([2, 5, 1], 3, 5) prints: 5  
    numbersBetween([2, 5, 1, 7, 4], 2, 6) prints: 2 5 4    '''
```

16. Modify the previous problem to print nothing, but put the selected numbers in a list, and return the list.

17. What is printed? .

```
x = 0 #1  
while x < 10: #2  
    x = 2*x + 1 #3  
    print(x, end=' ') #4
```

18. What is printed? .

```
s = 'Y' #1  
while len(s) < 3: #2  
    s = 2*s #3  
    print(s) #4
```

19. What is printed?

```
print(list(range(2, 5)))  
print(list(range(2, 14, 4)))
```

20. What is printed?

```
words = ['A', 'short', 'list'] #1  
print(len(words)) #2  
for s in words: #3  
    print(len(s)) #4
```

21. Write a function called **roll** that simulates  
one roll of a 6-sided die. Use the **random** module.  
The function should return a random integer  
between 1 and 6.

*Answers posted next Monday*