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 summary 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.

The review problems are several times as long as an exam.

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
```

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
```

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.

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. What is printed? .

```
x =  37                #1
while x > 7:           #2
    x = x - 2          #3
    print(x, end=' ')  #4
    if x > 12:         #5
        x = x - 10     #6
```

Answers on the next page

Exam 2 Review Problem Answers

1. HELLO
   MOM
2. HellothereMom (no spaces or new lines)

3a . C first two tests are false 5<3, 5>10, falls through to else
b. B first true part is 12 > 10. Never get to else
c. A stop at first test -2 < 3
d. C both tests false as in part a.

step by step – does not show the spaces and newlines, not a complete substitute for the final answer!

line	comment	line	comment	line	comment	line	comment
1	5 <3 false	1	12 <3 false	1	-2 <3 true	1	10 <3 false
3	5 > 10 false	3	12 > 10 true	2	print A	3	10 > 10 false
5	print C	4	print B			5	print C

4a. C b. AC c. BC d. A
Note the last if statement is completely separate from the part above, so the last test is always done. The middle test is only true if both comparisons are true. The last test is true if either comparison is true.

step by step – does not show the spaces and newlines, not a complete substitute for the final answer!

line comment

part a

1 5 == 3 false
3 5 <5 and 3>2: false and true: false
5 5>2 or 3>4: true or false: true
6 print C

part b

1 5 == 5 true
2 print A
5 5>2 or 5>4: true or true: true
6 print C

part c

1 1 == 5 false
3 1 <5 and 5>2: true and true: true
4 print B
5 1>2 or 5>4: false or true: true
6 print C

part d

1 1 == 1 true
2 print A
5 1>2 or 1>4: false or false: false

5. **1 3**
x is printed before being increased, so the first value is printed. The last value of x is 5, but x becomes that after the last time it is printed.

step by step – does not show the spaces and newlines, not a complete substitute for the final answer!

line	x	comment
1	1	
2		1<5 true: loop
3		print 1
4	3	
2		3<5 true: loop
3		print 3
4	5	
2		5<5 false: skip loop

6. 31 32 33
41 42 43

step by step – does not show the spaces and newlines, not a complete substitute for the final answer!

line	x	y	comment
1	30		first in outer list
2		1	first in inner list
3			30+1 = 31; print 31 (stay on line)
2		2	next in inner list
3			30+2 = 32; print 32 (stay on line)
2		3	last in inner list
3			30+3 = 33; print 33 (stay on line)
2		-	no more in list - done with inner loop
4			print (advance to new line)
1	40		next in outer list
2		1	start again with first inner list element
3			40+1 = 41; print 41 (stay on line)
2		2	next in inner list
3			40+2 = 42; print 42 (stay on line)
2		3	last in inner list
3			40+3 = 43; print 43 (stay on line)
2		-	no more in list - done with inner loop
4			print (advance to new line)
1	-		done with outer loop

7. a b aaa bbb

step by step – does not show the spaces and newlines, not a complete substitute for the final answer!

line	n	s	comment
1	1		first in outer list
2		a	first in inner list
3			print a (stay on line)
2		c	second in inner list
3			print c (stay on line)
2	.....	-..	no more in list - end inner loop
1	3		second in outer list
2		a	start again - first in inner list
3			print aaa (stay on line)
2		c	second in inner list
3			print ccc (stay on line)
2	.....	-..	no more in list - end inner loop
1			no more in list -- done with outer loop

8. **[0, 2, 4, 6]**

Remember square brackets and commas. Note range(4) is [0, 1, 2, 3].

line	nums	i	comment
1	[]		
2	[]	0	first in list
3	[0]	0	append 2*0 = 0
2	[0]	1	next in list
3	[0,2]	1	append 2*1 = 2
2	[0,2]	2	next in list
3	[0,2,4]	2	append 2*2 = 4
2	[0,2,4]	3	last in list
3	[0,2,4,6]	3	append 2*3 = 6
2	[0,2,4,6]	3	done with list
4	[0,2,4,6]	3	print [0, 2, 4, 6]

9.
```
# Creating a float safer:  Not clear if the number must be an int.
x = float(input("Enter a number: "))
    if x > 100:
      print("High")
    elif x < 50:
      print("Low")
    else:
      print("In between")
```

10.
```
c = Circle(Point(40, 50), 10)
c.draw(win)
```

11.
```
for num in numlist:
      print(2*num)
```

12.
```
for num in numlist:
    print("Twice {} is {}.".format(num,2*num)) #print final period; no space before
```

13.
```
for word in wordlist:
    print(word, end= ' ')
```

14.
```
lowVal <= num <= highVal
# lowVal <= num and num <= highVal  #alternate
```

15.
```
for num in numList:
    if lowVal <= num <= highVal:
        print(num, end = ' ')
```

16.
```
chosen = []
for num in numList:
    if lowVal <= num <= highVal:
        chosen.append(num)
return chosen
```

17. **1 3 7 15**

line	x	comment
1	0	
2	0	0 < 10 is True; loop'
3	1	2*0+1 = 1
4	1	print 1 (stay on same line)
2	1	1 < 10 is True; loop'
3	3	2*1+1 = 3
4	3	print 3 (stay on same line)
2	3	3 < 10 is True; loop'
3	7	2*3+1 = 7
4	7	print 7 (stay on same line)
2	7	7 < 10 is True; loop'
3	15	2*7+1 = 15
4	15	print 15 (stay on same line)
2	15	15 < 10 is False; skip loop

18. **YY**
YYYY

line	s	comment
1	Y	
2	Y	1 < 3 is True; loop'
3	YY	2*'Y' is 'YY'
4	YY	print YY
2	YY	2 < 3 is True; loop'
3	YYYY	2*'YY' is 'YYYY'
4	YYYY	print YYYY
2	YYYY	4 < 3 is False; skip loop

19. **[2, 3, 4]** # start with 2, end before 5
[2, 6, 10] # start with 2, end before 14, jumps of 4
They are list objects, hence the square brackets and commas when printed.

20. **3** First prints the length of the list (3 elements, each a string)
1 Next loop through each word, and print the length of each word:
5 Length for a string is measured by the number of characters.
4

21.
```
35 23 11 9 7
```

line	x	comment
1	37	
2		37 > 7 is true; loop
3	35	37-2=35
4		print 35 (all on same line)
5		35 > 12 is true
6	25	35-10=25
2		25 > 7 is true; loop
3	23	25-2=23
4		print 23 (all on same line)
5		23 > 12 is true
6	13	23-10=13
2		13 > 7 true; loop
3	11	13-2=11
4		print 11 (all on same line)
5		11 > 12 is false
2		11 > 7 true; loop
3	9	11-2=9
4		print 9 (all on same line)
5		9 > 12 is false
2		9 > 7 true; loop
3	7	9-2=7
4		print 7 (all on same line)
5		7 > 12 is false
2		7 > 7 false; done