Study Questions for 115 Final

Not guaranteed to be all inclusive, but good to work through.

I recommend highly that after you study the materials to try to answer these questions and similar questions to this (Evaluate your self). If you doubt your answer you can type the code into the python program and run.

1. What is the value of *x* while the function body is being evaluated?

```
def compute(x,y):
                a = x + 1
                b = y - 1
                return a * b

        j = 3
        k = 7
        result = compute(j,k)
```

   1. 7
   2. j
   3. 3
   4. k

2. What is the value of *y* while the function body is being evaluated?

```
def compute(x,y):
                a = x + 1
                b = y - 1
                return a * b
        j = 3
        k = 7
        result = compute(j,k)
```

   1. y
   2. 3
   3. k
   4. 7

3. What is printed by this program:

```
def almostSquare(x,y):
        (x - 1) * (y + 1)

print(almostSquare(5,5))
```

1. None because there is no explicit return
2. 25
3. nothing is printed because of an error
4. 24

5. What is printed by this program:

```
def almostSquare(x,y):
        (x - 1) * (y + 1)

x = almostSquare(5,5)
    print(x)
```

1. 25
2. nothing is printed because of an error
3. None because there is no explicit return
4. 24

6. By the conventions of this course, the first function your program should call is:
   1. the first one defined
   2. *start*
   3. *main*
   4. the last one defined

7. The # character means that:
   1. Python should ignore it and any remaining characters on the line
   2. Python should escape the following special character in a string
   3. Python should hash the following string into a number
   4. Python should treat the remainder of the line as a string

8. String methods: if str1 = " 987694 3290 " and st2 = "2"

a. len(st2) =
b. st2 + str1 =
c. str1 + st2 =
d. str1[5:9] =
e. str1.split("9") =
f. str1.strip() =
g. str1.replace('9','5') =

9. What does this code create in the variable x?

```
x = []
for i in range(3):
    x.append ( [i, i, i] )
```

a. x = [1,1,1,2,2,2,3,3,3]
b. x = [0,0,0,1,1,1,2,2,2,3,3,3]
c. x = [[0,0,0], [1,1,1], [2,2,2]]
d. x = [0, 0, 0, 1, 1, 1, 2, 2, 2]
e. x =[i, i, i]

10. what is the output of the following prints:

```
L = [4, 10, 8]
x = L.sort()
L.append(20)
L2 = L[1:]

print (x) _____________
print (L) _______________
print (L2)__________________________
```

11. What does each of the following expression evaluate to? Suppose that

L = ["These", "are", "a", "few", "words", "that", "we", "will", "use"].
(a) L[1:4]
(b) L[0:]
(c) L[len(L)-1]
(d) L[3:4]
(g) L[:len(L)-4]
(i) L[0] + L[1] + L[2]
(j) L[0:2] + L[4:6]

12. Is this function definition correct?

```
def pow(x,y)
    return x ** y
```

1. no, a colon character is missing
2. no, the return statement should not be indented
3. yes, it is correct as written
4. no, there should only be one formal parameter

13. For this problem, suppose that s is the string "Hello, how are you?" What does each of the following expression evaluate to?

(a) s.find("how") ____

(b) s.count("o") ____

(c) s.split()[1:3] ____

(d) s.upper() ____

(e) s.isalnum() ____

(f) s.replace("how", "who") ______________________________________

(g) s.split(",") ____

14. What's the output?

```
count = 0
loop = 0

while count < 4:
     loop = loop + 1
     count = count + 2

count = count + loop
print(count)
```

15. What's the output?

```
def h(x,y):
  ans = x * x + y * y
  return ans

def main():
  ans = 5
  result = h(3,4)
  print(ans, result)

main()
```

16. What's the output?

```
list1 = []
list2=[1,2,3]
list1.append("start")
list1.append(2)
list1.append(list2[0])
print(list1)
```

17. What's the output?

```
list2 = []
for i in range(2,5):
    list2.append(i)

for i in range(len(list2)):
    if i > 0:
        list2[i] = list2[i] + list2[i-1]

print(list2)
```

18. what is the output ?

```
x =4
y=0
try:
    z= x/y
    print (z)

except    ZeroDivisionError:
   print(“ red”)

 print(“white”)
```

19. Someone wrote the following code, intending to print **10 20 30 40**.

```
arr = [10, 20, 30, 40]
for index in range(1,5):
        print(arr[index])
```

What went wrong when it was run?

20. What's the output?

```
list1 = []
list2=[1,2,3]
list1.append("start")
list1.append(2)
list1.append(list2[0])
print(list1)
```

21. Compare and contrast lists and external files. The factors to consider include permanence, speed of access, direct/sequential access, size.

22. Show the contents of the list sample after the code segment is executed.

```
sample = []
for k in range(8):
    sample.append(10 - k)
```

23. Write a command to open a file for writing on your current folder with the name “myfile.txt” ?

24. What is the main difference between fileobject.readlines( ), fileObject.read( )

24. Write a function that would accept two integer parameters and return True if the two parameter values are equal of each other, and return False otherwise.

25. Write a function **prod** that has parameters of two lists **arr1** and **arr2** that are both the same size. This function should return the product of all components of arr2 by 2 for which the corresponding components of arr1 are negative. For example,

if arr1 were [-1, 4, 9, -2, 7]
if arr2 were [9, 5, 4, 2, 1]

The product returned would be 9 * 2 = 18.

26. Write a loop that will read in numbers from the keyboard until a 0 is entered, then report the largest and smallest numbers entered.

27. Write a function definition for a function called calc which has 3 integer parameters. The parameters represent two operands and a choice for operator. Your function should determine which operation is requested, perform the operation and return the result. The operation choices are: 1 for add, 2 for subtract, 3 for multiply and 4 for divide. The function should not allow division by zero; it should return 0 instead. The function does no input from the keyboard nor output to the screen. Your function must use only one return statement.

Example: calc(2, 4, 1) would return 6 because 2 + 4 = 6
calc(10, 15, 2) would return -5 because 10-15 = -5
calc (3, 9, 3) would return 27 because 3*9 = 27
calc (15, 4, 4) would return 3.75 because 15/4 = 3.75