

## Results

This assignment was comprised of 5 different sections with overarching themes including printing, operators, lists, dictionaries, and loops.

The printing section of this assignment was of an extremely basic form. A string was defined as `stringFirst` and was set equal to 'This is the first string.' The printed result was as expected, reading back what the string was set equal to. A second string labelled `stringStudent` was set equal to "I'm a student" and, when printed, read back "I'm a student." The last string the assignment asked to define posed a question about the students' feelings toward the class. Defined as `stringFeelings`, I printed the string, and thus the question. I furthered this section by setting a label, `x`, equal to three different responses on three different lines using a triple quotation mark, and printed the responses.

The operator section of this homework dealt with basic calculations and required a change in variable type during one portion. The results were the answers to  $(a+b)$ ,  $(a/b)$ , the integer portion of  $(a/b)$ , the remainder of  $(a/b)$ ,  $(a^b)$ , a boolean return for  $(a \neq b)$ , and a boolean return for  $(a > b)$ . Because `a=100` and `b=9` were interpreted as integers, the result of  $(a/b)$  automatically returns an integer value for the calculation instead of the integer and the following decimals. To fix this, the variable type of one of the values had to be changed. 'b' was changed to a float, which allowed the program to print the decimals as well. Returning the remainder of  $(a/b)$  was accomplished using the mod operator, denoted with the percent symbol (%).  $(a\%b)$  yields 1. '\*\*' is used in python for power functions.  $(a**b)$  is the same as the math expression  $(a^b)$ . For the first boolean operation, simply writing  $(a \neq b)$  returns the boolean value True

because 'a' does not equal 'b.' The second boolean operation was completed by writing (a>b), which returns the boolean value True because 'a' is greater than 'b.'

The list practice section required the making of lists, as well as appending and extending the set lists. The main list(List\_A) was comprised of integers, floats, and strings. The append function was used to create a nested list. A string was then inserted into the second place of the main list and then removed immediately after. The last item in the main list was then printed and removed. A new list was added that took everything from the third element(List\_A[2]) on and made it into its own list. That new list (List\_C) was multiplied by 2 in order to double its size, and was then reversed in sequence.

The dictionary practice part of the assignment only required the making of a given list, and the use of that list to detail how many of each unique digit were in the given list. This included the use of a for loop and the inclusion of an if else statement to add the elements into the dictionary. The physical code and output can be viewed in the appendix.

The final section was the loop practice section, which was certainly the most involved section of the homework. Three different functions, labelled sequence, InverseSequence, and OrderingList were defined for the requirements in this section. The goal was to check if the given list was an ascending sequential list or a descending sequential list and place a given number into the right spot in the given list. The OrderingList function defines what to do for any amount of elements in the given set and decides to either place the number in or call one of the other two previously mentioned functions. If one of the other two functions are called on, they then determine where to place the desired number.

## Appendix

#This is the print portion of the homework

```
stringFirst = 'This is the first string'
```

```
print(stringFirst)
```

```
stringStudent = "I'm a student"
```

```
print(stringStudent)
```

```
stringFeelings = 'How do you think of this course?'
```

```
x = """I am a little nervous about the course.
```

```
I am eager to learn python and hope I enjoy it.
```

```
I feel as though I am a bit behind the curve and need to catch up."""
```

```
print(x)
```

#This is the operator portion of the homework

```
a = 100
```

```
b = 9
```

```
c = a + b
```

```
print(c)
```

`print(a/float(b))`      # because a and b are both integers, the float type is used to print the decimals as well

`print(a/b)`      # because a and b are both integers, the division of a by b will be an integer value as well

`print(a%b)`      # % is the mod function which shows the remainder

`print(a**b)`      # \*\* is the power operator

`print(a != b)`

`print(a > b)`

#This is the list practice portion of the homework

`integer1 = 1`

`#print(type(integer1))`      #used to check to see if this is an integer, which python does recognize it as an integer

`integer2 = 2`

`#print(type(integer2))`      #used to check to see if this is an integer, which python does recognize it as an integer

`float1 = 10.1`

```
#print(type(float1))          #used to check to see if this is a float, which python does  
recognize it as a float
```

```
float2 = 10.2
```

```
#print(type(float2))          #used to check to see if this is a float, which python does  
recognize it as a float
```

```
string1 = 'This is item number 5'  #used to check to see if this is a string, which python does  
recognize it as a string
```

```
#print(type(string1))
```

```
string2 = 'This is item number 6'  #used to check to see if this is a string, which python does  
recognize it as a string
```

```
#print(type(string1))
```

```
List_A = [int(integer1), int(integer2), float(float1), float(float2), string1, string2]
```

```
List_B = [1, 2, 3, 4, 5]
```

```
List_A.append(List_B)
```

```
#print(List_A)                #used to check if List_B was added to List_A
```

```
List_A.insert(1, 'FE520')
```

```
#print(List_A)          #used to check if the string was added to List_A
```

```
List_A.remove('FE520')
```

```
#print(List_A)          #used to check if the string was added to List_A
```

```
print(List_A[6])
```

```
List_A.remove(List_A[6])
```

```
print(List_A)           #used to check if the last item was deleted
```

```
List_C = List_A[2: ]     # colon is used to show that we are including the 3rd element to  
the end of the list
```

```
print(List_C)
```

```
List_C.extend(List_C)
```

```
#print(List_C)          #used to check if the list was doubled
```

```
List_C.reverse()
```

```
#print(List_C)          #used to check if the sequence was reversed
```

```
#This is the practice dictionary portion of the homework
```

```
A = [1, 2, 3, 5, 10, 1, 4, 10, 11, 20, 50, 100]
```

```
count = {}
```

```
for item in A:
```

```
    if (item in count):
```

```
        count[item] += 1
```

```
    else:
```

```
        count[item] = 1
```

```
print(count)
```

```
#This is the loop condition practice portion of the homework
```

```
List = [1, 2, 4, 9, 17, 25, 63]
```

```
List2 = [63, 25, 17, 9, 4, 2, 1]
```

```
def sequence(lst, number):
```

```
    temp = 0
```

```
    for i in range(len(lst)):
```

```
        if(number < lst[i]):
```

```
            break
```

```
temp+=1
```

```
return lst[0: temp] + [number] + lst[temp :]
```

```
print(sequence(List, 13))
```

```
def InverseSequence(lst, number):
```

```
    temp = 0
```

```
    for i in range(len(lst)):
```

```
        if(number > lst[i]):
```

```
            break
```

```
    temp+=1
```

```
    return lst[0: temp] + [number] + lst[temp :]
```

```
print(InverseSequence(List2, 13))
```

```
def OrderingList(lst, number):
```

```
    if (len(lst) == 0):
```

```
        return lst + [number]
```

```
    if (len(lst) == 1):
```

```
        if(lst[0] > number):
```

```
            return [number] + lst[0]
```



```
else:
```

```
    return [lst[0]] + [number]
```

```
else:
```

```
    if(lst[0] > lst[(len(lst)-1)]):
```

```
        return InverseSequence(lst,number)
```

```
else:
```

```
    return sequence(lst,number)
```

```
#Lst1 = []
```

```
#Lst2 = [1,3]
```

```
#Lst3 = [3,1]
```

```
#Lst4 = [2]
```

```
#print(OrderingList(Lst1,2))      # This whole section was used to check and see if all of the  
funcitons work properly
```

```
#print(OrderingList(Lst2,2))
```

```
#print(OrderingList(Lst3,2))
```

```
#print(OrderingList(Lst4,2))
```

```
#I pledge my honor that I have abided by the Stevens Honor System.
```

```
#Dominic Zecchino
```

```
1 #This is the print portion of the homework
2
3 stringFirst = 'This is the first string'
4 print(stringFirst)
5
6 stringStudent = "I'm a student"
7 print(stringStudent)
8
9 stringFeelings = 'How do you think of this course?'
10 x = """I am a little nervous about the course.
11 I am eager to learn python and hope I enjoy it.
12 I feel as though I am a bit behind the curve and need to catch up."""
13 print(x)
14
15 #This is the operator portion of the homework
16
17 a = 100
18 b = 9
19
20 c = a + b
21 print(c)
22 print(a/float(b)) # because a and b are both integers, the float type is used to print the decimals as well
23 print(a/b) # because a and b are both integers, the division of a by b will be an integer value as well
24 print(a%b) # % is the mod function which shows the remainder
25 print(a**b) # ** is the power operator
26
27 print(a != b)
28 print(a > b)
29
30 #This is the list practice portion of the homework
31
32 integer1 = 1
33 #print(type(integer1)) #used to check to see if this is an integer, which python does recognize it as an integer
34
35 integer2 = 2
36 #print(type(integer2)) #used to check to see if this is an integer, which python does recognize it as an integer
37
38 float1 = 10.1
39 #print(type(float1)) #used to check to see if this is a float, which python does recognize it as a float
40
41 float2 = 10.2
42 #print(type(float2)) #used to check to see if this is a float, which python does recognize it as a float
43
44 string1 = 'This is item number 5' #used to check to see if this is a string, which python does recognize it as a string
45 #print(type(string1))
46
47 string2 = 'This is item number 6' #used to check to see if this is a string, which python does recognize it as a string
48 #print(type(string2))
49
50 List_A = [int(integer1), int(integer2), float(float1), float(float2), string1, string2]
51 List_B = [1, 2, 3, 4, 5]
52
53 List_A.append(List_B)
54 #print(List_A) #used to check if List_B was added to List_A
55
56 List_A.insert(1, 'FE520')
57 #print(List_A) #used to check if the string was added to List_A
58
59 List_A.remove('FE520')
60 #print(List_A) #used to check if the string was added to List_A
61
62 print(List_A[6])
63 List_A.remove(List_A[6])
64 print(List_A) #used to check if the last item was deleted
65
```

```
64 print(List_A)           #used to check if the last item was deleted
65
66 List_C = List_A[2: ]     # colon is used to show that we are including the 3rd element to the end of the list
67 print(List_C)
68
69 List_C.extend(List_C)
70 #print(List_C)           #used to check if the list was doubled
71
72 List_C.reverse()
73 #print(List_C)           #used to check if the sequence was reversed
74
75 #This is the practice dictionary portion of the homework
76
77 A = [1, 2, 3, 5, 10, 1, 4, 10, 11, 20, 50, 100]
78
79 count = {}
80 for item in A:
81     if (item in count):
82         count[item] += 1
83     else:
84         count[item] = 1
85 print(count)
86
87 #This is the loop condition practice portion of the homework
88
89 List = [1, 2, 4, 9, 17, 25, 63]
90 List2 = [63, 25, 17, 9, 4, 2, 1]
91
92
93
94 def sequence(lst, number):
95
96     temp = 0
97     for i in range(len(lst)):
98         if(number < lst[i]):
99             break
100         temp+=1
101
102     return lst[0: temp] + [number] + lst[temp :]
103
104 print(sequence(List, 13))
105
106 def InverseSequence(lst, number):
107     temp = 0
108     for i in range(len(lst)):
109         if(number > lst[i]):
110             break
111         temp+=1
112     return lst[0: temp] + [number] + lst[temp :]
113
114 print(InverseSequence(List2, 13))
115
116 def OrderingList(lst, number):
117     if (len(lst) == 0):
118         return lst + [number]
119     if (len(lst) == 1):
120         if(lst[0] > number):
121             return [number] + lst[0]
122         else:
123             return [lst[0]] + [number]
124     else:
125         if(lst[0] > lst[(len(lst)-1)]):
126             return InverseSequence(lst,number)
127         else:
128             return sequence(lst, number)
```

```

116 def OrderingList(lst, number):
117     if (len(lst) == 0):
118         return lst + [number]
119     if (len(lst) == 1):
120         if(lst[0] > number):
121             return [number] + lst[0]
122         else:
123             return [lst[0]] + [number]
124     else:
125         if(lst[0] > lst[(len(lst)-1)]):
126             return InverseSequence(lst,number)
127         else:
128             return sequence(lst,number)
129
130 #Lst1 = []
131 #Lst2 = [1,3]
132 #Lst3 = [3,1]
133 #Lst4 = [2]
134 #print(OrderingList(Lst1,2))           # This whole section was used to check and see if all of the funcitons work properly
135 #print(OrderingList(Lst2,2))
136 #print(OrderingList(Lst3,2))
137 #print(OrderingList(Lst4,2))
138
139 #I pledge my honor that I have abided by the Stevens Honor System.
140 #Dominic Zecchino
141

```

I pledge my honor that I have abided by the Stevens Honor System.

Dominic Zecchino

