From Homework for CSC 160

Problem discovered in zybooks, Lab 3.12.

Code submitted:

my\_flower1 = input()

my\_flower2 = input()

my\_flower3 = input()

your\_flower1 = input()

your\_flower2 = input()

their\_flower = input()

*# 1. TODO: Define my\_list containing my\_flower1, my\_flower2, and my\_flower3*

*# in that order*

my\_list = [my\_flower1, my\_flower2, my\_flower3]

# in that order

*# 2. TODO: Define your\_list containing your\_flower1 and your\_flower2*

*# in that order*

your\_list = [your\_flower1, your\_flower2]

# in that order

*# 3. TODO: Define our\_list by concatenating my\_list and your\_list*

our\_list = my\_list + your\_list

print(our\_list)

*# 4. TODO: Append their\_flower to the end of our\_list*

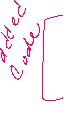
our\_list.append(their\_flower)



print(our\_list)

print("my\_flower2", id(my\_flower2), "their\_flower", id(their\_flower))

print("id my\_flower2:", our\_list.index(my\_flower2),"id their\_flower", our\_list.index(their\_flower))



print(my\_flower2, their\_flower)

# 5. TODO: Replace my\_flower2 in our\_list with their\_flower

for i in range(len(our\_list)):

if our\_list[i] == my\_flower2:

our\_list[i] = their\_flower

print(our\_list)

print("id index [1]", id(our\_list[1]),"id index[3]", id(our\_list[3]))



# 6. TODO: Remove the first occurrence of their\_flower from

# our\_list without using index()

our\_list.remove(their\_flower)

print(our\_list)

# 7. TODO: Remove the second element of our\_list

our\_list.pop(1)

print(our\_list)

Problem

Table

Description automatically generatedWhen running this code, it runs normally until it encounters the third of 5 iterations of inputs. The other 4 results were as expected. The code on #3 work correctly until it reaches:



**# 5. TODO: Replace my\_flower2 in our\_list with their\_flower**

When this happens, my\_flower2 should be replaced with their\_flower at it’s indexed location. Adding some additional code, which is highlighted in yellow on the previous page, I determined that my\_flower2 was located at index 1 or the second position on the 2nd row. It should have been replaced with tulip which is index 5 or in the 6th position on the same column. But what happens is that their\_flower gets split into two identical identities and gets placed in both index 1 and index 3 on the 3rd row.

When zybooks is performing properly, their\_flower located in column 2 takes on the indexed position of my\_flower2 in row 3.



Table

Description automatically generated



However, when we get to the 3rd iteration. Something interesting happens. their\_flower which is located at index 2 in row 2 gets copied to two location in row 3 (at index 1 and 3, both which have the same unique object ID). This obviously causes a knock-on effect for the rest of the code.

Table, calendar

Description automatically generated

