CS5001: Lab 7. Due on Friday, Oct-27-2023.

Name(s): Xujia Qin

Email(s): qin.xuj@northeastern.edu

You can work on this lab either individually or in small group of two or three students. If working in a group, include names of all the students in the submission PDF.

Getting credit for this lab. This lab handout has several empty boxes that prompt you to answer a question. As part of the lab, you are to write the answers to these questions inside the boxes/blanks. When you are finished, you should create a PDF and upload it on Canvas. If you don't finish, you have until 11:59 PM on Friday, Oct-27-2023 to submit.

What computer to use? If your primary computer is a laptop, bring it to the lab to work on, as lab is an excellent opportunity to get started with Python on your machine. You should follow the instructions on the course website. Ask a TA for help if you have problems with your installation. If you prefer, you could also use one of the machines in the lab room to work on this lab assignment.

Lab Materials. Lab materials can always be found on Canvas under the appropriate lab posting.

For today's lab, you need this handout which is also online on Canvas.

1 Lists: Basic Ideas

These exercises feature lists of numbers.

1.1 Typical list errors

Remember, it is square brackets with comma separators for setting up short lists, e.g., [10,20,30]. For each of these examples, how does Python respond? **(15 points, 5 each)**

1.1.1)

```
>>> x = [10 20 30]
```

```
  File "/Users/cathyqin/Desktop/lab7.py", line 1
    x = [10 20 30]
         ^^^^^
SyntaxError: invalid syntax. Perhaps you forgot a comma?
```

1.1.2)

```
>>> x = [10, 20, 30]
>>> x[3] = 40
```

```
Traceback (most recent call last):
  File "/Users/cathyqin/Desktop/lab7.py", line 2, in <module>
    x[3] = 40
    ~^^^
IndexError: list assignment index out of range
```

1.1.3)

```
>>> x = [10 ; 20]
```

```
  File "/Users/cathyqin/Desktop/lab7.py", line 1
    x = [10 ; 20]
            ^
SyntaxError: invalid syntax
```

1.2 Lists are a Type (10 points, 5 each)

1.2.1)

What does Python say after this:

```
>>> x = [1,2,3]
>>> type(x)
```

```
    <class 'list'>
```

1.2.2) What does Python say after this:

```
>>> x = [10]
>>>  y  =  10
 >>> type(x)         <class 'list'>
>>> type(y)          <class 'int'>
```

1.3 The Void Methods append and extend

Do these problems without the computer. Then check your answers by using interactive Python.

1.3.1) What do you think Python would say after the following? **(2 points)**

```
>>> x = [10,20,30]
>>> x.append(99)
>>>  x
```

```
[10,20,30,99]
```

Now, run the above code using interactive Python mode. What does Python say? **(2 points)**

```
[10,20,30,99]
```

1.3.2) What do you think Python would say after the following? **(2 points)**

```
>>> x = [10,20,30]
>>> y = x.append(99)
 >>> type(y)
```

```
<class 'list'>
```

Noted, because append is a void function, it will not return anything, instead it will manipulate the list to add the element, so it will return nonetype.

Now, run the above code using interactive Python mode. What does Python say? **(2 points)**

```
<class 'NoneType'>
```

1.3.3) What do you think Python would say after the following? **(2 points)**

```
>>> x = [10,20,30]
>>> y = [40,50]
 >>> x.extend(y)
>>>  x
```

```
[10,20,30,40,50]
```

Now, run the above code using interactive Python mode. What does Python say? **(2 points)**

```
[10,20,30,40,50]
```

1.3.4) What do you think Python would say after the following? **(2 points)**

```
>>> x = [10,20,30]
>>> y = [40]
>>> x.extend(y)
>>>  x
```

```
[10,20,30,40]
```

Now, run the above code using interactive Python mode. What does Python say? **(2 points)**

```
[10,20,30,40]
```

1.3.5) What do you think Python would say after the following? **(2 points)**

```
>>> x = [10,20,30]
>>>  y  =  40
>>> x.extend(y)
```

error to add the non-same type?

Now, run the above code using interactive Python mode. What does Python say? **(2 points)**

```
Traceback (most recent call last):
  File "<pyshell#32>", line 1, in <module>
    x.extend(y)
TypeError: 'int' object is not iterable
```

1.4 The Void Method insert

Do these problems without the computer. Then check your answers by using interactive Python.

1.4.1) What do you think Python would say after the following? **(1 points)**

```
>>> x = [1,2,3,4]
>>>  i  =  3
>>>  y  =  99
>>> x.insert(i,y)
 >>>  x
```

[1,2,3,99]

Noted, insert method inserts the element at the specified index and pushes the existing elements to the right.

Now, run the above code using interactive Python mode. What does Python say? **(2 points)**

```
[1, 2, 3, 99, 4]
```

1.4.2) What do you think Python would say after the following? **(2 points)**

```
>>> x = [1,2,3,4]
>>>  i  =  3
>>> y = [10,20]
>>> x.insert(i,y)
>>>  x
```

(A list can have a list as an entry.)

[1,2,3,[10,20],4]

Now, run the above code using interactive Python mode. What does Python say? **(2 points)**

```
[1, 2, 3, [10, 20], 4]
```

1.5 The Void Method sort

Do these problems without the computer. Then check your answers by using interactive Python.

1.5.1) What do you think Python would say after the following? **(2 points)**

```
>>> x = [4,1,3,2]
>>> x.sort()
>>>  x
```

[1,2,3,4]

Now, run the above code using interactive Python mode. What does Python say? **(2 points)**

```
[1, 2, 3, 4]
```

1.5.2) What do you think Python would say after the following? **(2 points)**

```
>>> x = [4,1,3,2]
>>> x = x.sort()
>>> print(x)
```

[1, 2, 3, 4]

Noted, because the sort is a void function and does not return a new list. Therefore, it returns None.

Now, run the above code using interactive Python mode. What does Python say? **(2 points)**

```
None
```

1.5.3) What do you think Python would say after the following? **(2 points)**

```
>>> x = [4,1,3,2]
>>> x = x.sort(reverse=True)
>>>  x
```

[4,3,2,1]

Noted, sort is a void function and returns nothing.
When you assign this return to x, the result will come out nothing.

Now, run the above code using interactive Python mode. What does Python say? **(2 points)**

It comes out nothing.

1.6 The Fruitful Methods count and pop

Do these problems without the computer. Then check your answers by using interactive Python.

1.6.1) What do you think Python would say after the following? **(3 points)**

```
>>> x = [4,1,3,2,4]
>>> m = x.count(4)
>>>  m
```

2

Now, run the above code using interactive Python mode. What does Python say? **(3 points)**

2

1.6.2) What do you think Python would say after the following? **(3 points)**

```
>>> x = [10,20,30,40]
>>> m = x.pop(1)
>>> print (m,x)
```

```
20 [10,30,40]
```

Now, run the above code using interactive Python mode. What does Python say? **(3 points)**

```
20 [10, 30, 40]
```

1.6.3) What do you think Python would say after the following? **(3 points)**

```
>>> x = [10,20,30]
>>> a = x.pop(0)
>>> b = x.pop(0)
>>> c = x.pop(0)
>>> print (a,b,c,x)
```

```
10 20 30 []
```

Now, run the above code using interactive Python mode. What does Python say? **(3 points)**

```
10 20 30 []
```

1.6.4) What do you think Python would say after the following? **(3 points)**

```
>>> x = [10,20,30]
>>> a = x.pop(len(x)-1)
>>> b = x.pop(len(x)-1)
>>> c = x.pop(len(x)-1)
 >>> print (a,b,c,x)
```

```
30 20 10 []
```

Now, run the above code using interactive Python mode. What does Python say? **(3 points)**

```
30 20 10 []
```

1.7 Slicing and Subscripts

Do these problems without the computer. Then check your answers by using interactive Python.

1.7.1) What do you think Python would say after the following? **(3 points)**

```
>>> x = [3,2,0,1]
>>> y = x[x[x[1]]]
>>>  y
```

```
3
```

Now, run the above code using interactive Python mode. What does Python say? **(3 points)**

```
3
```

1.7.2) What do you think Python would say after the following? **(3 points)**

```
>>> x = [3,2,0,1]
>>> y = x[1:]
>>> y.append(x[0])
>>>  y
```

```
[2,0,1,3]
```

Now, run the above code using interactive Python mode. What does Python say? **(3 points)**

```
[2, 0, 1, 3]
```