CMSC 201 Section 40

Spring 2020

Lab 7 – Working with 2D Lists

Value: 10 points

Release date: March 30, 2020

Due date: April 2, 2020 before midnight

Purpose:

This lab is set up to give you practice working with 2-dimensional (2D) lists. A 2D list is a list that is made up of one-dimensional lists. You will demonstrate the ability to create a 2D list, access an individual element of a 2D list; access a row of a 2D list; access a column of a 2D list; and add a row to a 2D list.

Assignment:

Start with the following table, which is the first part of one of the data files from Project 1:

Name	StudentID	Project1	Project2	Project3	Test1	Test2	Test3
Leonard	A12345	80	80	64	75	123	189
Sheldon	B34093	78	72	63	75	125	200
Amy	C39080	45	80	64	70	67	165
Wil	D49830	63	62	77	72	114	199
Bernadette	E38909	78	78	80	68	89	187

Step 0: Create the file for this lab

Create a file called ‘lab7.py’ which you will submit with this lab.

Step 1: Creating a 2D list

Write code that creates this table as a 2D list. That is, write a statement like the following:

```
grade_table = [

        [‘Name’, ‘StudentID’, ‘Project1’, ‘Project2’,’Project3’, ‘Test1’, ‘Test2’, ‘Test3’},

        [‘Leonard’,’A12345’,80,80,64,75,123,189],

        [‘Sheldon’,’B34093’,78,72,63,75,125,200],

# now add the other three rows here

   ]
```

Verify that you have entered the table correctly by printing it.

Step 2: Accessing an individual element of a 2D list

You will next access individual elements in the 2D list. Remember that an element is accessed by giving the list name, then the row number in square brackets [], then the column number in square brackets []. THE ROW ALWAYS COMES FIRST!!!

Remember also that we always start counting from 0. So the first column in the first row is grade_list[0][0], NOT grade_list[1][1].

Write print statements to print the following list elements:

a. The third column in the first row
b. The first column in the third row
c. The last column in the last row

Step 3: Accessing a row of a 2D list

The row number always comes first in referencing a 2D list. If you have a 2D list and you only give one subscript, it is interpreted as the row number and you get the entire row – the list that makes up that element of the 2D list.

grade_list[0] gives you the header row – that is, grade_list[0] is [‘Name’, ‘StudentID’, ‘Project1’, ‘Project2’,’Project3’, ‘Test1’, ‘Test2’, ‘Test3’}

Write a print statement that prints out the row that starts with ‘Sheldon.’

Step 4: Appending a row to a 2D list

Since you can access an entire row at a time, you can append or insert a row by using the methods you learned for one-dimensional lists.

If you want to append the next row in the table, you can simply write

```
        grade_list.append([‘Mary’,’F13047’,23,79,78,69,100,184])
```

and it will add the new row to the bottom of the 2D list.

Append the row that starts ‘Mary’ now.

Then insert the following row as the new third row (row number 2!!!) in your list:

```
[‘Beverly’, ‘A09324’, 80,71,79,54, 125, 172]
```

Print the value of len(grade_list)

Step 5: Accessing a column of a 2D list

While entire rows can be accessed at once, unfortunately there is no way in Python to access a column all at once. If you want to access each element in column 3, you have to use a loop – either for or while – to go through loop one row at a time and access that element. To print the elements in column 3, you would do something like:

```
for j in range(len(grade_list)):
        print(grade_list[j][3])
```

If you want to insert a new column 2, containing students’ majors, you would similarly use a loop. Suppose you had the majors, with a header element, defined in the list

```
majors = [‘Major’, ‘Physics’, ‘Psychology’, ’Chemistry’, ’Neuroscience’, ‘Drama’, ‘Biology’,’Finance’]
```

You would insert that as column 2 by:

```
 for j in range(len(grade_list)):
        grade_list[j].insert(2,majors[j])
```

Insert the column into your 2D list by running the for loop above.

Now insert a new column 3 containing lab grades, which are contained in the list

```
lab_grades = [‘Labs’, 100,92, 87. 79, 96, 100, 100]
```

Submittal

Submit your lab 7 as follows:

```
submit cmsc201 lab7 lab7.py
```

You have until midnight Thursday, April 2 to submit this lab for full credit.