\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**CSC121 PYTHON Programming**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

LAB 06 **LISTS [PART 2]**

# Objectives

In this lab assignment, students will learn:

- How to use for statements to iterate over lists

- How to use the range function

- How to write code to copy and concatenate lists

- How to create and use nested lists

- How to create and use tuples

- How to use list comprehensions to generate lists

# Goals

In this lab assignment, students will demonstrate the abilities to:

- Use for statements to iterate over lists

- Use the range function

- write code to copy and concatenate lists

- Create and use nested lists

- Create and use tuples

- Use list comprehensions to generate lists

# Instruction and Problems

Write a Python program for each of the problems in this lab. Please use PyCharm to type and test your programs. Submit the Python files to Blackboard for credit. In this lab, you should submit 4 Python files, one for each problem.

## Problem 1

A teacher wants a program to give extra points to students who fail a test. Write a Python program to do the following:

1. Ask the user to enter 5 test scores. Store the scores in a list. Display the list.
2. Copy all 5 test scores to another list. Use a loop to examine each test score in the new list. If the score is below 60, add 10 extra points to the score. Display the list.
3. Compare the old score and new score of each student. If the old score and new score are different, display the two scores.

The following is an example.

Enter a test score: 45

Enter a test score: 77

Enter a test score: 88

Enter a test score: 52

Enter a test score: 90

All scores: [45.0, 77.0, 88.0, 52.0, 90.0]

Students who scored below 60 get 10 extra points.

All scores: [55.0, 77.0, 88.0, 62.0, 90.0]

Students whose scores have changed:

Old score: 45.0 New score: 55.0

Old score: 52.0 New score: 62.0

Save your Python program in a file named **Lab06P1.py**. Submit the file to Blackboard for credit.

## Problem 2

Write a Python program to do the following:

1. Use a for loop and a random integer generator to generate 10 random integers in 1 through 15. Store the random integers in a tuple. Display the tuple. [Hint: you may want to store the random integers in a list first and then convert the list to a tuple]
2. Create a new tuple. Copy the first three elements of the tuple in part (a) to this tuple. Display this tuple.
3. Create a new tuple. Copy the last three elements of the tuple in part (a) to this tuple. Display this tuple.
4. Concatenate the two tuples in part (b) and part (c). Display the concatenated tuple.
5. Sort the concatenated tuple. Display the sorted tuple.

The following is an example. There is no user input in this program.

Tuple of 10 random numbers: (12, 3, 4, 7, 5, 9, 7, 3, 1, 7)

Tuple of first 3 numbers: (12, 3, 4)

Tuple of last 3 numbers: (3, 1, 7)

Two tuples concatenated: (12, 3, 4, 3, 1, 7)

Two tuples concatenated and sorted: (1, 3, 3, 4, 7, 12)

Save your Python program in a file named **Lab06P2.py**. Submit the file to Blackboard for credit.

## Problem 3

Three figure skaters, Jean, Kayla and Connie, compete in a meet. Each skater receives 4 scores. Write a program to do the following.

1. Ask the user to enter 4 scores for Jean. Store the scores in a list. Display the list.
2. Ask the user to enter 4 scores for Kayla. Store the scores in a list. Display the list.
3. Ask the user to enter 4 scores for Connie. Store the scores in a list. Display the list.
4. Create a list to store the three score lists. This is a list of lists that contains three elements: Jean’s score list, Kayla’s score list, Connie’s score list. Display this list of lists.
5. Use a set of nested for loops to add 1 extra point to every score of every skater in the list of lists created in part (d).
6. Display the modified list of lists. Each score should be 1 point higher than before.
7. For each skater in the list of lists, sort her 4 scores in ascending order.
8. Display the modified list of lists. Each skater’s 4 scores should be in ascending order.

The following is an example.

Please enter Jean's scores one by one.

Enter a score: 77

Enter a score: 74

Enter a score: 75

Enter a score: 72

Jean's scores: [77.0, 74.0, 75.0, 72.0]

Please enter Kayla's scores one by one.

Enter a score: 81

Enter a score: 77

Enter a score: 80

Enter a score: 79

Kayla's scores: [81.0, 77.0, 80.0, 79.0]

Please enter Connie's scores one by one.

Enter a score: 69

Enter a score: 74

Enter a score: 72

Enter a score: 70

Connie's scores: [69.0, 74.0, 72.0, 70.0]

All scores: [[77.0, 74.0, 75.0, 72.0], [81.0, 77.0, 80.0, 79.0], [69.0, 74.0, 72.0, 70.0]]

All scores after extra point: [[78.0, 75.0, 76.0, 73.0], [82.0, 78.0, 81.0, 80.0], [70.0, 75.0, 73.0, 71.0]]

All scores after sorting: [[73.0, 75.0, 76.0, 78.0], [78.0, 80.0, 81.0, 82.0], [70.0, 71.0, 73.0, 75.0]]

Save your Python program in a file named **Lab06P3.py**. Submit the file to Blackboard for credit.

## Problem 4

This program is about list comprehension. We are going to use list comprehension to generate sequences of powers.

1. Use list comprehension to generate the first, second, third and fourth powers of 2. Store the results in a list. The first power of 2 is 21 = 2. The second power of 2 is 22 = 4. The third power of 2 is 23 = 8. The fourth power of 2 is 24 = 16. Our goal is to generate the following list:

[21, 22, 23, 24]

which equals

[2, 4, 8, 16]

In Python, we use n\*\*p to calculate the pth power of n. For example, 2\*\*1 is 21, 2\*\*2, is 22, 2\*\*3 is 23, and 2\*\*4 is 24.

You need an input sequence when you use list comprehension to generate the list of powers of 2. You are required to use a range function to create the input sequence. Display the list of powers of 2 after it is generated.

1. Use list comprehension to generate the first, second, third and fourth power of 3. Store the result in a list. The first power of 3 is 31 = 3. The second power of 3 is 32 = 9. The third power of 3 is 33 = 27. The fourth power of 3 is 34 = 81. Our goal is to generate the following list:

[31, 32, 33, 34]

which equals

[3, 9, 27, 81]

You need an input sequence when you use list comprehension to generate the list of powers of 3. You are required to use a range function to create the input sequence. Display the list of powers of 3 after it is generated.

1. In mathematics, a matrix is a rectangular array of values arranged in rows and columns. In Python, we can store a matrix in a list of lists. Each list in the list of lists is a row of the matrix. For example, the list of lists [ [2, 4, 8, 16], [3, 9, 27, 81] ] represents a matrix of two rows and four columns. The list [2, 4, 8, 16] is the first row, while the list [3, 9, 27, 81] is the second row.

Use nested list comprehension to generate the following list of lists:

[ [2, 4, 8, 16],

[3, 9, 27, 81],

[4, 16, 64, 256],

[5, 25, 125, 625],

[6, 36, 216, 1296] ]

The first row is a list of powers of 2. The second row is a list of powers of 3. The third row is a list of powers of 4. The fourth row is a list of powers of 5. The fifth row is a list of powers of 6.

You need two input sequences when you use nested list comprehension to generate the lists of powers shown above. You are required to use two range functions to create the two input sequences. Display the list of lists after it is generated. There is no need to display each row in a separate line. Just use one print statement to display everything.

The following is the expected output.

Powers of 2: [2, 4, 8, 16]

Powers of 3: [3, 9, 27, 81]

Matrix: [[2, 4, 8, 16], [3, 9, 27, 81], [4, 16, 64, 256], [5, 25, 125, 625], [6, 36, 216, 1296]]

Save your Python program in a file named **Lab06P4.py**. Submit the file to Blackboard for credit.

# Grading rubric for Problem 1

Creating list of original scores [5 points]

Creating list of modified scores [10 points]

Displaying pairs of different scores [10 points]

# Grading rubric for Problem 2

Creating tuple of random integers [5 points]

Copying elements of tuple [5 points]

Concatenating tuples [5 points]

Sorting tuple [10 points]

# Grading rubric for Problem 3

Creating lists of scores [5 points]

Creating list of lists [5 points]

Adding extra point to scores in list of lists [5 points]

Sorting scores in list of lists [5 points]

# Grading rubric for Problem 4

Using list comprehension to generate powers of 2 [8 points]

Using list comprehension to generate powers of 3 [8 points]

Using list comprehension to generate lists of powers [14 points]