ITSE 1359 – Lab 4 Assignment (Refer to Ch. 3 and 4 as needed):

NOTE: Lab 4 requires you to work on and submit two programs. Read this document carefully and comply with submission and grading criteria. Your first problem (Drop Lowest Grade) is more prescriptive (use video provided), while the second program (Rainfall Stats) is more open ended (solve as specified).

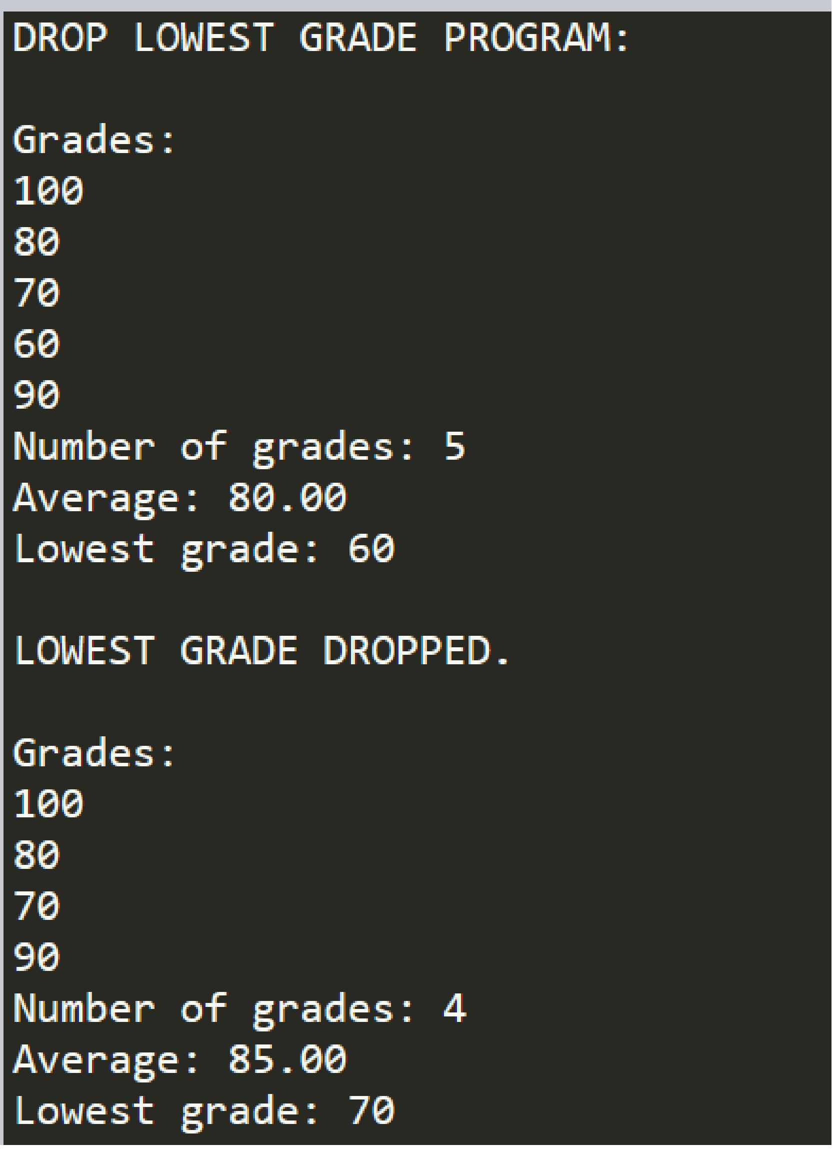
Problem #1 - Drop Lowest Grade (drop_grade.py):

Problem Definition: Create a Python program that drops the lowest grade in a list. Call your file drop_grade.py. Create a list of five grades. Use this test data in this order: 100, 80, 70, 60, 90. Show the original grades followed by the number of grades, average, and lowest grade. Drop the lowest grade and then show the grades again followed by the new number of grades, average, and lowest grade. Your output must match screenshot provided. Do not forget general and specific comments. Keep your solution in a safe place because you will need it again in another lab.

Pseudocode / Comments (Drop Lowest Grade):

```
# print report name – DROP LOWEST GRADE PROGRAM
# create list of 5 grades and initialize
# show original grades using for in Loop
# use sum function on grades to calculate the total
# calculate the average by dividing total by len(grades)
# print number of grades using len function
# print average formatted to 2 decimal places
# find the lowest grade using min function and print lowest grade
# get the index of the lowest grade
# use del statement to delete the lowest grade by passing index
# print LOWEST GRADE DROPPED.
# show grades after dropping lowest grade using for in Loop
# use sum function to calculate the total after dropping lowest grade
# compute new average
# print new number of grades
# print new average
# find new lowest grade and print
```

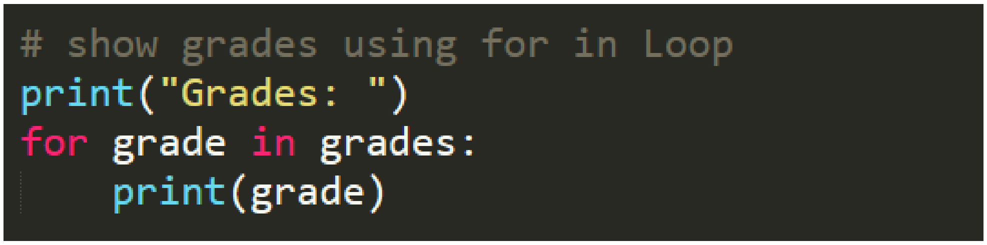
Screenshot of Output (Drop Lowest Grade):



```
DROP LOWEST GRADE PROGRAM:

Grades:
100
80
70
60
90
Number of grades: 5
Average: 80.00
Lowest grade: 60

LOWEST GRADE DROPPED.

Grades:
100
80
70
90
Number of grades: 4
Average: 85.00
Lowest grade: 70
```

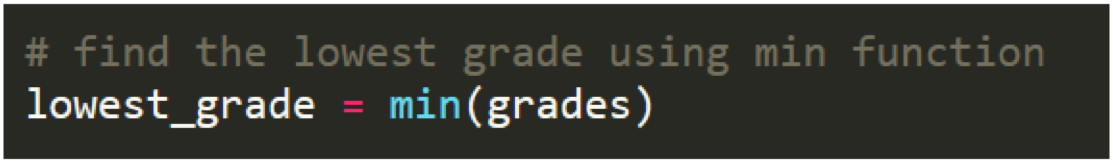
Code Examples (Drop Lowest Grade):

How to show grades in list using for in Loop?



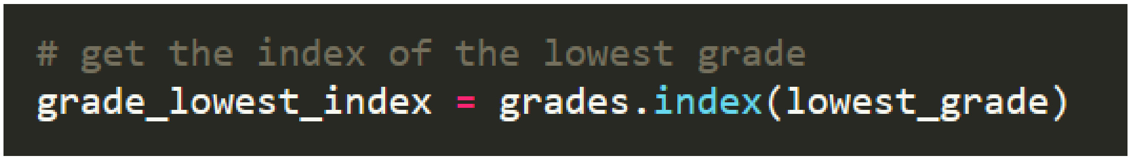
```python
# show grades using for in Loop
print("Grades: ")
for grade in grades:
    print(grade)
```

How to find the lowest grade?



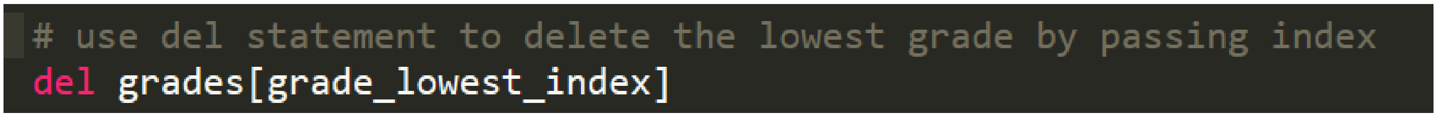
```python
# find the lowest grade using min function
lowest_grade = min(grades)
```

How to get the index of lowest grade? (Not in book)



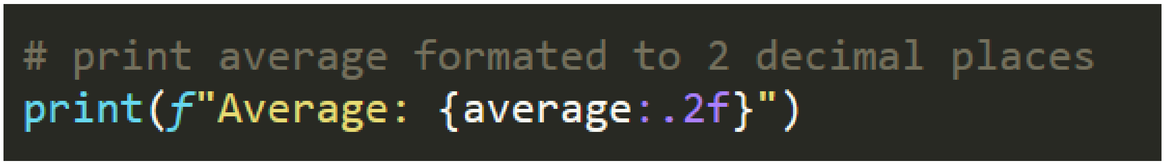
```python
# get the index of the lowest grade
grade_lowest_index = grades.index(lowest_grade)
```

How to delete the lowest grade?



```python
# use del statement to delete the lowest grade by passing index
del grades[grade_lowest_index]
```

How to format average? (formatting not in book – covered in previous labs)



```python
# print average formated to 2 decimal places
print(f"Average: {average:.2f}")
```

Another way?

Many times, there is another way of doing something. Above we found the index of the lowest value and then we used that index with the del statement. This may be easier (one line – instead of two):

```
# remove lowest grade
grades.remove(lowest_grade)
```

So now we can delete these two statements:

```
# get the index of the lowest grade
grade_lowest_index = grades.index(lowest_grade)

# use del statement to delete the lowest grade by passing index
del grades[grade_lowest_index]
```

Point: If you can do it an easier way and still have your code be easy to understand then it's a win/win.

What is difference between del and remove? >>> CSEstack.org

Program #2 – Rainfall Stats (rainfall_stats.py):

Problem Definition: Create a Python program that shows average monthly rainfall in inches and some summary statistics. Call your program rainfall_stats.py. Do not forget general and specific comments.

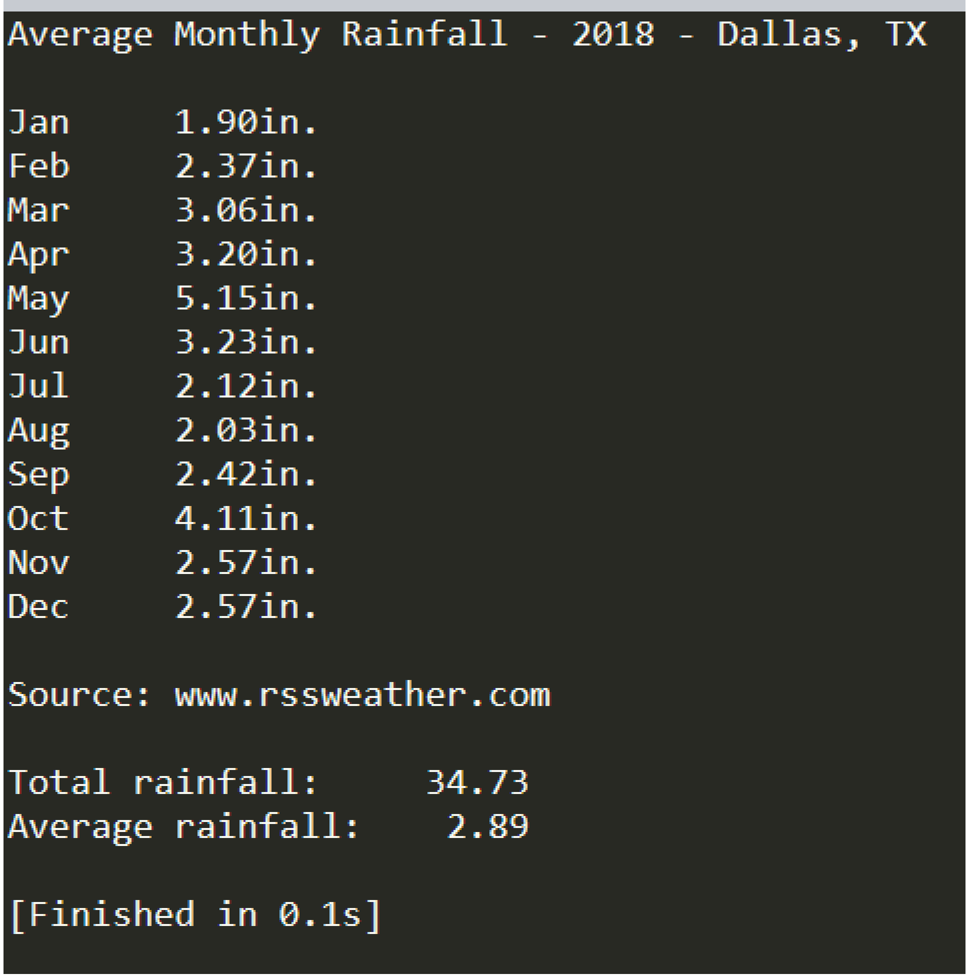
Creating the parallel lists: Initialize a **string list** called "months" to hold the months of a year and list them in lower case and in abbreviated format (3 char max). Initialize a **numeric list** called "inches" that corresponds to the average rainfall in inches for each month. *Use the data in the screenshot provided to populate both lists.*

Looping the lists: Your program should use the range() function to loop the lists. Your program should also provide basic rain statistics: total and average. Instead of manually showing the data for each month by writing 12 print statements, use a for loop with the range() function to loop through both lists to show the data formatted in two columns.

After you show the data from the two lists, calculate, and show some basic stats: total rainfall and average rainfall. Make sure you use formatted strings to show output as shown in the screenshot provided.

Output: Your output should be in columnar format and be formatted as shown in the screen shot below (numbers formatted to 2 decimal places). Stats should also appear formatted as shown (numbers should be right aligned). You can use white space to simulate the columns or use the escape code for tab.

Screenshot of Output (Rainfall Stats):



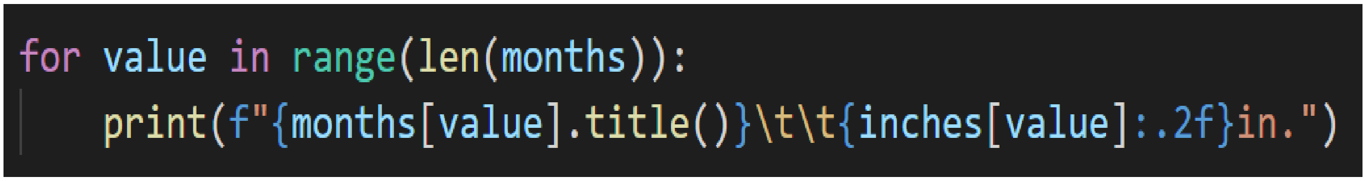
```
Average Monthly Rainfall - 2018 - Dallas, TX

Jan     1.90in.
Feb     2.37in.
Mar     3.06in.
Apr     3.20in.
May     5.15in.
Jun     3.23in.
Jul     2.12in.
Aug     2.03in.
Sep     2.42in.
Oct     4.11in.
Nov     2.57in.
Dec     2.57in.

Source: www.rssweather.com

Total rainfall:     34.73
Average rainfall:    2.89

[Finished in 0.1s]
```

Pseudocode / Comments (Rainfall Stats):

```
# initialize string list of the months
# initialize numeric list with average monthly rainfall

# print report header – Average Monthly Rainfall
# use for loop with the range function – pass len of month list
    # print month and average rainfall by accessing temporary variable

# print credit line showing source of data
# use sum function to find the total rainfall and assign to total variable
# dividing total by len function of list and assign to average variable
# display total and average rainfall for year
```

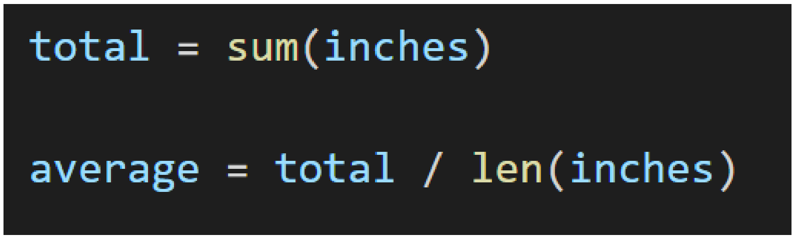
Code Examples (Rainfall Stats):

How to use a for loop to traverse lists:



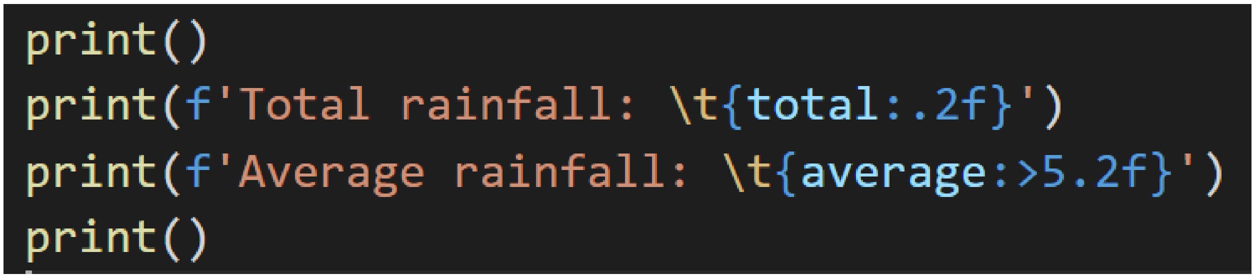
```python
for value in range(len(months)):
    print(f"{months[value].title()}\t\t{inches[value]:.2f}in.")
```

How to use sum() function and find average:



```python
total = sum(inches)

average = total / len(inches)
```

How to format output (note average rainfall is left aligned):



```
print()
print(f'Total rainfall: \t{total:.2f}')
print(f'Average rainfall: \t{average:>5.2f}')
print()
```

Food for Thought:

This lab asks you to write a lot of code that is not well organized. Later, you will learn about functions and how you can make your program more modular and capable of code reuse. Hang on we will cover that soon!

Submit your lab assignment:

Using the Canvas assignment tool, upload your completed work (2 files) to the lab 4 assignment. Attach the first file (drop_grade.py) and then the second (rainfall_stats.py) and then submit.

All lab assignments must be submitted using the CANVAS assignments tool. Lab Assignments will not be accepted any other way**.** Make sure you submit your work to the right lab assignment number otherwise you will not get credit.

VIP – **Keep your Drop Lowest Grade Solution**: Keep your solution to this program in a safe place because you will need it again for another lab assignment.

Grading Criteria:

- ✓ You must use the provided video to help you do Drop Lowest Grade.
- ✓ Don’t forget general comments.
- ✓ Specific comments are optional.
- ✓ Use white space to make your code easy to read.
- ✓ Comply with PEP-8 conventions for variable names, file names, etc.
- ✓ Satisfy the problem definition and other grading standards.
- ✓ Your work should not have syntax errors.
- ✓ Your work must be your own.
- ✓ Match your output screen to screenshot provided.
- ✓ Any deviations from lab specifications will result in points off.
- ✓ If you want to vary, do so on your own.

VIP Videos:

How to do the drop lowest grade program (step-by-step)?

Lab 4 Drop Lowest Grade Video

Lectures and other videos: VIEW MY PYTHON PLAYLIST

Closing:

If you have questions about this lab, send me a message using canvas inbox or attend zoom office hours. See Unit 0 for link for my office hours. Programming tutors are also available and listed in your Canvas class as announcements.

By-the-way, a great way to get ready for your lab assignments (and exams) is do the Try It Yourself problems in your book. Most of the solutions are on the authors website and I also discuss them in my YouTube video lectures.

Another way to get ready for your lab assignments is to review my lecture notes for each chapter. In Unit 0, find the link for my chapter notes.

Warning: Labs will increase in complexity with each lab assignment. A beginner may need 5 to 15 hours to complete each lab. Please start early because there will be no extensions.

Consider doing this lab over a three-day period:

- ✓ Day 1 – Get confused and run out of time.
- ✓ Day 2 – Research and debug errors.
- ✓ Day 3 – Polish, double-check everything, and submit.

Have an exception free day!

Prof. Benavides