

**Lab 4****Learning Objectives**

- Develop Python programs that generate sequences.
- Develop Python programs that use the accumulator pattern.
- Begin working with the textbook author's graphics library.

**Activities**

Complete the following activities in the provided Python file. Make sure to include meaningful comments, input prompts, and output messages to receive full credit.

**1. Sequences, series, and products**

Read through the attached PDF on generating sequences, series, and products. Then, complete the following functions in the starter file so that they compute the first  $n$  elements of the given sequence, series, or product. For a sequence, display the first  $n$  terms. For a series or product, display the sum/product of the first  $n$  terms/factors. Pattern A has been done for you. You can run your solutions individually, or all together using the provided `patterns()` function.

- 2 4 6 8 10 ...
- $4 + 9 + 14 + 19 + 24 + \dots$
- 0 4 0 4 0 4 ...
- $3 - 6 + 6 - 9 + 9 - 12 + 12 - \dots$
- $\frac{4}{1} * \frac{8}{3} * \frac{16}{5} * \frac{32}{7} * \frac{64}{9} * \dots$

**2. Calculating your grade**

You want to determine your average homework grade. Write a function, `hw_stats()`, that asks the user for a number of scores to be entered. The function should ask for each score and then output the mean (average), rounded to one digit. For example:

```
Enter the number of assignments: 3
Enter your grade on HW1: 85
Enter your grade on HW2: 93.3
Enter your grade on HW3: 90.5
Your overall average is 89.6
```

Each input prompt should include the assignment number (HW1, HW2, HW3, ...).

### 3. Calculating a standard deviation

The [standard deviation](#) of a dataset is a measurement of how spread out it is. Modify your `hw_stats()` function to also display the standard deviation of your homework scores, using the following formula:

$$SD = \sqrt{\frac{1}{n} \sum_{i=0}^{n-1} (scores[i] - mean)^2}$$

(If you're unfamiliar with the capital sigma  $\Sigma$ , [you can read about summation notation here](#).)

Since this formula requires the mean that you computed in Exercise 2, naturally any calculations you need to perform must occur after your code for the previous exercise. Here is a possible strategy for solving this problem (you may choose another approach):

1. Modify your original solution to also store all the homework scores in a list.
2. After calculating the mean, loop through the list of scores and accumulate the summation part of the formula:  $\sum_{i=0}^{n-1} (scores[i] - mean)^2$
3. Divide by  $n$ , take the square root, and display the result.

You can test your answer using Python's built-in `stdev()` function in the `statistics` library (don't forget to import it!). If you have a list of numbers called `values`, you can display its standard deviation like this:

```
print(statistics.stdev(values))
```

### 4. Getting started with graphics

Read through the provided function `draw_shapes()`, which opens a blank window where the user can click to draw and move a circle. Try running the function to make sure you understand how it works. Then, modify it to display a 20 pixel square instead of a circle. The square should still be centered on the location the user clicks.

Upload `lab4.py` to the OAKS dropbox before the deadline. Make sure you have most of the exercises completed before your lab meeting.