**FOCUS TOPICS**

- Manipulating lists
- Iteration with for
- Using the random module

**TASK**

In this Lab we will illustrate the use of looping through a sequence with an example of record-keeping of 10 quiz scores for a grade book. Instead of working with scores that would be entered by the user, we will simulate the scores by generating random numbers.

We will then implement a grading policy of replacing the lowest score with the highest score, and calculate averages for the scores before and after the change. Then we will take a random sample of the scores, sort them, and shuffle them into a random order.

Finally, we will iterate over the list to print the scores one by one.

**INSTRUCTIONS:**

1. Create a new CoLab notebook called yourname_Lab5.

2. Use the import statement to make the `random` module available to our code.

3. Create an empty list called _scores_ that will later hold individual integer values for 10 quiz scores.

4. Use a `for` loop to create a loop that will iterate 10 times.
   a. Use the `range` function to create an iterator that will be used by the `for` loop.
   b. In the body, use the `random` module's `randint` function to get a random number between 0 and 100 inclusive.
   c. Append the generated random number to the end of the _scores_ list.

5. Calculate and print some statistics about the series. See the sample output for the format.
   a. Print the quiz scores as a list (not individually).
   b. Calculate and print the average score.
   c. Calculate and print the lowest score.
   d. Calculate and print the highest score.

6. Replace the lowest score with the highest score and print information about the replacement.
   a. Use the appropriate list method to find the position of the list item with the lowest score (if there are multiple lowest scores, use only the first one).

b. Use an assignment to store the highest score at the lowest score's location.

c. Print a line that describes the replacement of the lowest score (see output for format).

7. Calculate and print the average score value after the replacement took place.

   a. Print the new quiz scores as a list (not individually).

   b. Calculate and print the average score.

8. Use the `random` module's `choice` method to pick a random score from the list and print it.

9. Use the `random` module's `sample` method to pick a sample of 3 scores from the list and print them.

10. Use the list's `sort` method to sort the *scores* from lowest to highest and print them.

11. Use the `random` module's `shuffle` method to randomly shuffle the *scores* and print them.

12. Use a **for** loop to iterate over the list of scores. In each iteration, print the individual score and its distance from the average of all the scores (computed in step 7). See the Sample output for the formatting.

13. Check your output carefully to ensure that it matches the sample output. For the purposes of this lab, "matching" means that all spacing, wording, and punctuation in your output should match the sample output exactly for the format and text labels. Since the numbers are generated randomly, your numbers will be different from the Sample output.

**SAMPLE OUTPUT:**

This output uses random numbers so your results will differ. Use this sample as a reference for the wording and format.

```
Before replacement: [69, 62, 42, 12, 14, 20, 75, 67, 44, 91]
Average score: 49.6
Lowest score: 12
Highest score: 91

Replaced the lowest score of 12 at position 3 with the highest score of 91.
After replacement:  [69, 62, 42, 91, 14, 20, 75, 67, 44, 91]
Average score: 57.5

Random score: 69
Random sample: [44, 91, 69]
Sorted: [14, 20, 42, 44, 62, 67, 69, 75, 91, 91]
Shuffled: [91, 67, 75, 62, 44, 91, 69, 20, 14, 42]

Individual scores and their dispersion:
91 : distance from average: 33.5
67 : distance from average: 9.5
75 : distance from average: 17.5
62 : distance from average: 4.5
44 : distance from average: -13.5
91 : distance from average: 33.5
69 : distance from average: 11.5
20 : distance from average: -37.5
14 : distance from average: -43.5
42 : distance from average: -15.5
```