Understanding Indentation and Code Blocks in Python

Tutorial 4 - Exercise 3

Write a program that will allow a user to enter a number of scores until -9 is entered. When -9 is entered, print the average of the scores entered. Use a while loop. Ensure that at least one score has been entered before calculating the average (division by zero would produce an error).

Contents:

Step 1: understand what the user requires:

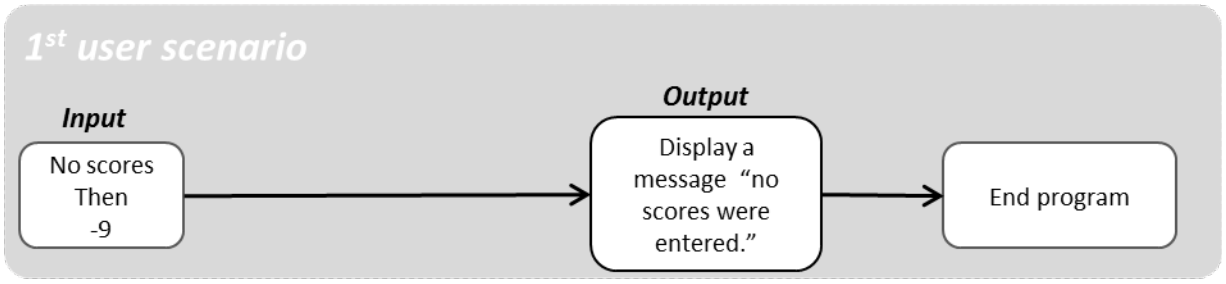
- The user wants to key in numbers (scores) from the keyboard.
- The user can decide to exit the program by keying in -9 from the keyboard.
- The user wants to keep typing numbers (scores) repeatedly until he/she decides to exit (-9)
- Once the user decided to exit (keyed in -9)
  - The user wants the program to calculate the average of all numbers entered by him/her from the keyboard.
  - The average is calculated by

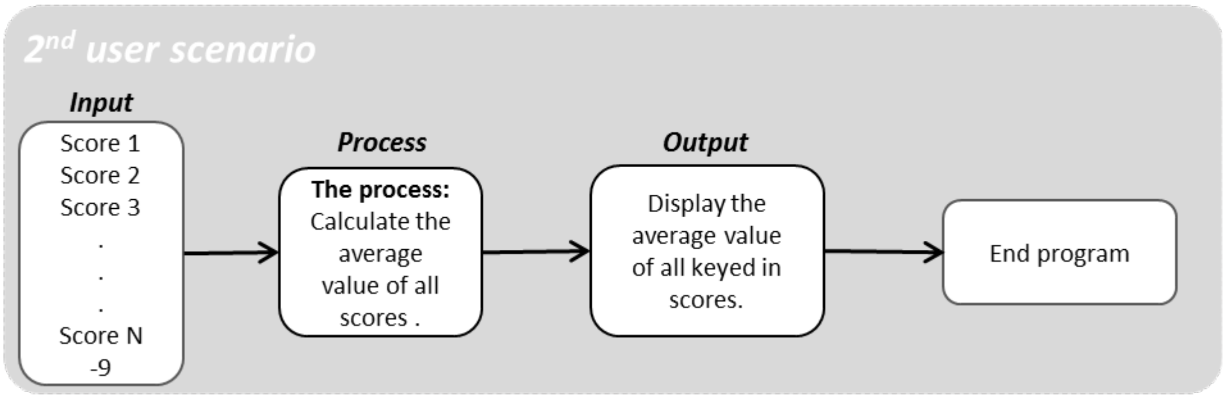
$$\text{Average of scores} = \frac{\text{score1} + \text{score2} + \text{score3} + \cdots + \text{scoreN}}{\text{Number of entered scores}}$$

  - The user wants to see the average of scores displayed on the screen.
  - In case the user does not key in any scores, the user doesn't want to get an error as a result of dividing by Zero when calculating the average.

Step 2: understand what the program requires:

- Understand the inputs to the program and the outputs (results) and the process (calculation) between both.
- The inputs are the typed-in numbers by the user from the keyboard, the output is the average values of all numbers, and the process is the calculations of the average value of all typed in numbers.




2nd user scenario
Input
Score 1
Score 2
Score 3
.
.
.
Score N
-9
Process
The process: Calculate the average value of all scores .
Output
Display the average value of all keyed in scores.
End program

- Understand the program flow before the inputs are entered (initiation phase) and when the inputs are given (the process) and when the user wishes to exit the loop (terminate the loop).

"The process is a calculation of average values of scores after they are entered from the keyboard, and the user decided to exit the process."

Initiating the process:

Each independent instruction or a set of instructions is written in a standalone block of code. The process needs two key values to start the calculation of average, the number of scores entered in a variable called *count*, and the sum of all entered scores in a variable called *total*.

Before the user has entered any scores from the keyboard, both key values *count* and *total* have no value, so we assign them a zero value. We call this an initiation phase. Then the user should enter the first *score* from the keyboard.

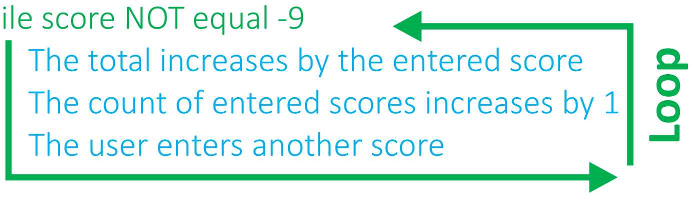
Block 1: total = 0
Block 2: count = 0
Block 3: score = User enters a score from the keyboard

During the Process

This exercise wants the user to be able to enter the first score then the second then the third and so on which indicates there is a repetition in the user action (this is a loop). If the user is allowed a set number of scores, then we use a *for loop* otherwise if the user has the freedom to type as many scores as needed then the use of *while loop* is appropriate. A *while loop* must have a value which causes it to break otherwise it will run forever, this value is called a **sentinel value**.

The loop will also have to keep accumulating (adding) all the entered scores hence the *total* will have to keep increasing by the value of the new *score* entered. Every time a new score is entered, the *count* of scores is increasing by 1

```
Block 4: While score NOT equal -9
            The total increases by the entered score
            The count of entered scores increases by 1
            The user enters another score
```



Note that each program code after the *while* statement is equally indented because the running of these instructions are dependent on the entered *score* meeting the Sentinel value (score NOT equal -9)

Terminating the loop

To get the result (average of scores) the user must stop entering scores (exit), but this action can't be performed because of the continuing running loop. First, the loop has to be broken, exiting the loop is called termination. Breaking the loop can be done by violating the loop term (score NOT equal -9), so by entering (-9), we will break the loop and then moves to a new block of code for new instruction.

Validating the user input

Before getting the final result (average), test if the user entered a score. Dividing by zero number of scores to get the average will cause an error.

For this validation, we could use a conditional statement such as *if statement*.

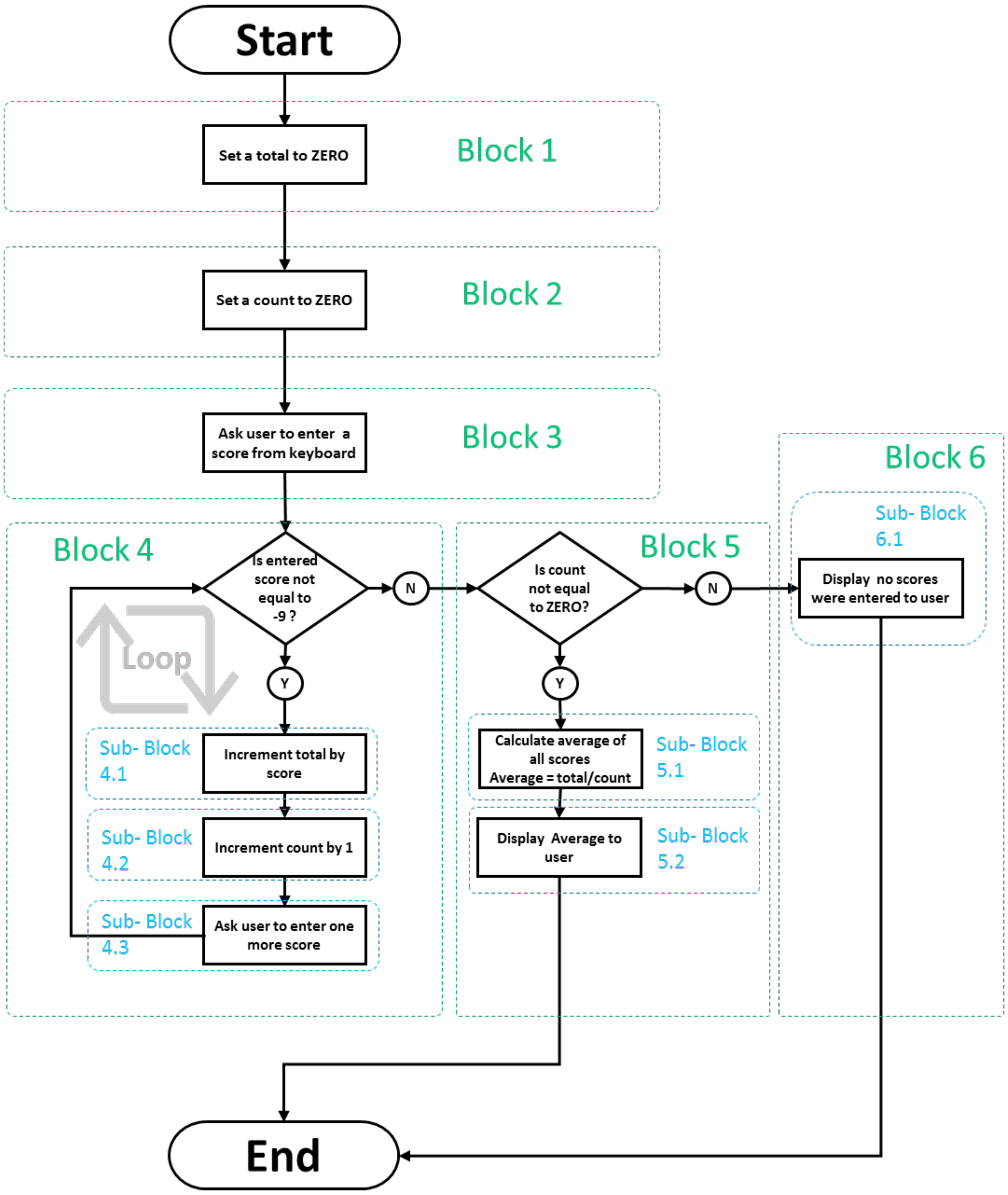
```
Block 5: if the user entered a score (count NOT equal Zero)
            Calculate the average value of entered scores
            Display the calculated average to the user on the screen
```

✓ OR ✗

Note that each program code after the *if statement* is equally indented because the running of these instructions are dependent on meeting the conditional value (count NOT equal -9)
Conditions can also be violated by choosing any other value which does not comply with the test value (count NOT equal Zero), so violating a condition results in moving to a new block of independent code.

```
Block 6: else
            Display a message to the user "No scores were entered."
```

The flow chart



The program code

There are different ways to write programs, the following solution in this paper is an example of one way to solve this exercise to understand the importance of indentation in python.

Independent Blocks are not indented

```
# Program Starts

total = 0    # Block 1: Set a total to ZERO

count = 0  # Block 2: Set a count to ZERO

# Block 3: Ask user to enter  a score from keyboard
score = int(input( "Enter score, (Enter -9 to end): " ) )

# Block 4: Is the entered score not equal to -9?
while score != -9:

        #Sub-block 4.1: Increment total by score
        total = total + score

        #Sub-block 4.2: Increment count by 1 >> Sub-block 4.2
        count = count + 1

        #Sub-block 4.3:Ask user to enter one more score
        score = int(input( "Enter score, (Enter -9 to end): " ) )

# Block 5: Is count not equal to ZERO (division by zero generates an error)
if count != 0: # division by zero would be a run-time error

        #Sub-block 5.1: Calculate average of all entered scores
        average = float( total ) / count

        #Sub-block 5.2: Display  Average to user
        print ("Class average is", average)

# Block 6: User did not enter a score so count is zero
else:

        #Sub-block 6.1: Display  no scores were entered to user
        print ("No scores were entered")

# Program Ends
```

Sub-blocks are indented (Dependent on While value)

Sub-blocks are indented (Dependent on count ≠ 0)

indented