

CS 112, Foundations of CS

Lab 3: Loops

Submit to Canvas

Computer
Science

This lab is worth 100 points. The goal for this lab is:

I. Getting familiar with looping structures in Python

If you do not complete the lab in the time allotted, then please return to the lab in your spare time, and complete it by the **due date, which is specified on Canvas**.

Preliminaries

- For this lab, create another folder, called **lab3**.
- **USE SPYDER TO EDIT CODE -- PYTHON3 ONLY**

I. Average Exam Score Calculator

For this programming task you will make a program which calculates the average of the exam scores. The exam scores will be inputs from the user. You will use a while loop to populate a list of exam scores with the function **input()**. Then you will print out the exam scores. After that you will add all of the exam scores and find the average of them.

1. To get started create a new python file (mouse-click on the *File* menu, then select *New file*). If you need help please ask or refer to lab 1.
2. In the comment section on top, include your name, today's date and program name (Lab 3, AverageExamScoresCalculator.py).
3. Create two variables, "**sum_elements**" and "**average_score**". Assign 0 to both of them. You will use them later on.
4. Create an empty list and name it "**scores**". An empty list would just be the square brackets. For example **cats = []**.
5. In this step you will populate the list with the use of the **input()** function, inside of a **while loop**. We want to set up the while loop, in a way that lets the user input as many, or as little exam scores as they want. To this we can have a conditional for the while loop be something that will always evaluate to true. Inside of the while loop we can have an if-else statement, checking if input is **-1**. If the input from the user is **-1**, then we can use a break statement to exit out of the while loop. To get started with your while loop type the following:

while 1 > 0:

A conditional "**1 > 0**", will make an infinite loop, and the only way to exit is by using a break

statement.

Inside of the while loop get users input by using the input function with the message “**Enter an exam score or -1 to stop:** ”. Store users input into a variable named “**user_input**”. Remember you are expecting integer values, you should cast it to an **int**.

6. Inside of the while loop, which you wrote for step #5, write an if- else. Which checks if “**user_input**” is equal to -1, If it is exit out of the while loop with a **break** statement. And in the else part add “**user_input**” to the list “**scores**”. This can be done with the append method:

scores.append(user_input)

7. Save and run your code. If there aren’t any errors then you will be prompted to enter exam scores one by one, until you enter -1. If there are errors, do fix them before moving on.
8. Invoke the print function with the message “**The exam scores:**”.
9. Write a for loop which goes through the list of scores and prints them out. If you add the parameter **end=**”, ” to the print function, it will print out the exam scores on one line.
10. In this step you will write another for loop. The for loop iterates through the list of “**scores**”, and adds all of the exam scores into the variable “**sum_elements**”. This can be done a few different ways, an easy one would be to use a for loop with “**range(len(scores))**”. Inside of the for loop update “**sum_elements**” by “**sum_elements + scores[i]**”. “**i**” being the variable name used by the for loop to represent items from the list.
11. Calculate the average exam score with : **sum_elements / len(scores)** .Store the result in the variable “**average_score**”.
12. Invoke two print statements. The first print statement should be an empty one, simply type “**print()**”. This is to have a line break from the previous print statement, used to print out the exam scores. The second print statement should print out a message and the average exam score.
13. Sample output of the program can be seen below. Run your program a few times, trying out different exam scores.

```

Enter an exam score or -1 to stop : 87
Enter an exam score or -1 to stop : 78
Enter an exam score or -1 to stop : 56
Enter an exam score or -1 to stop : 98
Enter an exam score or -1 to stop : 95
Enter an exam score or -1 to stop : 93
Enter an exam score or -1 to stop : 70
Enter an exam score or -1 to stop : 82
Enter an exam score or -1 to stop : -1
The exam scores :
87, 78, 56, 98, 95, 93, 70, 82,
Average of the exam scores : 82.375

Enter an exam score or -1 to stop : 89
Enter an exam score or -1 to stop : 12
Enter an exam score or -1 to stop : 67
Enter an exam score or -1 to stop : 99
Enter an exam score or -1 to stop : 100
Enter an exam score or -1 to stop : -1
The exam scores :
89, 12, 67, 99, 100,
Average of the exam scores : 73.4

```

Rubric

Upload your source code to Canvas. Here's what we are looking for, when grading your submission. In later lab submissions, there will be additional items that will be graded, which will be described in future lectures.

The .py file must be thoroughly commented. If your code breaks (crashes) because you've been unable to fix a syntax error, then the comments will allow you to receive partial credit.

For this lab, make sure that the following file is uploaded to Canvas:

AverageExamScoresCalculator.py

File / task	Points
I. AverageExamScoresCalculator.py	100
Total	100