



Lab 8 – Loops II

Chapter:	5. Loops
Time:	80 Minutes

Lab 8

Objectives

- To write programs for executing statements repeatedly by using *while* and *for* loops.
- To develop loops following the loop design strategy.
- To control a loop with a sentinel value.
- To use *for* loops to implement counter-controlled loops.
- To write nested loops.
- To implement program control with *break* and *continue*.
- To learn the techniques for minimizing numerical errors.
- To learn loops from a variety of examples.

Current Lab Learning Outcomes (LLO)

By completion of the lab the students should be able to

- Use *while* and *for* loops.
- Write counter-controlled loops.
- Write sentinel-controlled loops.
- Avoid infinite loops.
- Use *break* and *continue* keywords to control loops.
- Write nested loops.

Lab Requirements

- PyCharm (IDE).




Practice Activities with Lab Instructor (20 minutes)

Problem 1

Programming Exercises (5.10)


Write a program that prompts the user to enter the number of students and each student's score, and displays the highest score.

Here are some sample runs:



```
Enter the number of students: 5 <enter>
Enter a student name: Ahmad <enter>
Enter a student score: 95.2 <enter>
Enter a student name: Omar <enter>
Enter a student score: 93.5 <enter>
Enter a student name: Jamal <enter>
Enter a student score: 95.5 <enter>
Enter a student name: Yeaser <enter>
Enter a student score: 80.5 <enter>
Enter a student name: Bander <enter>
Enter a student score: 66 <enter>

Top student Jamal's score is 95.5
```



```
Enter the number of students: 2 <enter>
Enter a student name: Ahmad <enter>
Enter a student score: 95.2 <enter>
Enter a student name: Omar <enter>
Enter a student score: 93.5 <enter>

Top student Ahmad's score is 95.2
```

Solution

Phase 1: Problem-Solving Phase:

See comments on the code below.

Phase 2: Implementation Phase:

1. Create a new project and name it "Lab 8".
2. Create a new file and name it "activity_1.py".
3. Write the following code in the file:

activity_1.py

```
1 # Prompt the user to enter the number of students
2 numOfStudents = eval(input("Enter the number of students: "))
3
4 # Initialize the variables for the highest score
5 student1 = input("Enter a student name: ")
6 score1 = eval(input("Enter a student score: "))
7
8 # Get students scores
9 for i in range(numOfStudents - 1):
10     student = input("Enter a student name: ")
11     score = eval(input("Enter a student score: "))
12
13     # Check the score of the entered student with the highest score
14     if score > score1:
15         student1 = student
16         score1 = score
17
18 # Display the output
19 print() # Empty line
20 print("Top student " + student1 + "'s score is " + str(score1))
```

Problem 2

Programming Exercises (5.23)

Write a program that lets the user enter the loan amount and loan period in number of years and displays the monthly and total payments for each interest rate starting from 5% to 8%, with an increment of 1/8.

The following are formulas that you may need to solve the problem:

$$\text{monthlyInterestRate} = \frac{\text{annualInterestRate}}{1200}$$

$$\text{monthlyPayment} = \frac{\text{loanAmount} \times \text{monthlyInterestRate}}{1 - \frac{1}{(1 + \text{monthlyInterestRate})^{\text{numberOfYears} \times 12}}}$$

$$\text{totalPayment} = \text{monthlyPayment} \times \text{numberOfYears} \times 12$$

Here is a sample run:

```
Enter loan amount, for example 120000.95: 10000 <enter>
Enter number of years as an integer, for example 5: 1 <enter>
```

Interest Rate	Monthly Payment	Total Payment
5.000 %	856.07	10272.90
5.125 %	856.65	10279.77
5.250 %	857.22	10286.65
5.375 %	857.79	10293.53
5.500 %	858.37	10300.41
5.625 %	858.94	10307.30
...		
7.375 %	867.00	10403.97
7.500 %	867.57	10410.89
7.625 %	868.15	10417.82
7.750 %	868.73	10424.75
7.875 %	869.31	10431.68
8.000 %	869.88	10438.61

Solution

Phase 1: Problem-Solving Phase:

See comments on the code below.

Phase 2: Implementation Phase:

1. Open the project "Lab 8" if it was not opened or create it if it was not existing.
2. Create a new file and name it "activity_2.py".
3. Write the following code in the file:

activity_2.py

```
1 # Enter loan amount
2 loanAmount = eval(input(
3     "Enter loan amount, for example 120000.95: "))
4
5 # Enter number of years
6 numOfYear = eval(
7     input("Enter number of years as an integer, for example 5: "))
8
9 # Display the header
10 print() # Empty line
11 print(format("Interest Rate", "<20s"), end= "")
12 print(format("Monthly Payment", "<20s"), end= "")
13 print(format("Total Payment", "<20s"))
14
15 annualInterestRate = 5
16 while annualInterestRate <= 8:
17     # Obtain monthly interest rate
18     monthlyInterestRate = annualInterestRate / 1200
19
20     # Compute the monthly Payment and total Payment
21     monthlyPayment = loanAmount * monthlyInterestRate / \
22         (1 - (pow(1 / (1 + monthlyInterestRate), numOfYear * 12)))
23     totalPayment = monthlyPayment * numOfYear * 12
24
25     # Display results
26     print(format(annualInterestRate, ">5.3f"), "%",
27         format(monthlyPayment, "20.2f"), format(totalPayment, "20.2f"))
28     annualInterestRate += 1 / 8
```




Individual Activities (60 minutes)

Problem 3

Programming Exercises (5.11)

Write a program that prompts the user to enter the number of students and each student's score, and displays the highest and second highest scores.

Here is a sample run:



```
Enter the number of students: 5 <enter>
Enter a student name: Ahmad <enter>
Enter a student score: 95.2 <enter>
Enter a student name: Omar <enter>
Enter a student score: 93.5 <enter>
Enter a student name: Jamal <enter>
Enter a student score: 95.5 <enter>
Enter a student name: Yeaser <enter>
Enter a student score: 80.5 <enter>
Enter a student name: Bander <enter>
Enter a student score: 66 <enter>


Top two students:
Jamal's score is 95.5
Ahmad's score is 95.2
```

Problem 4

Programming Exercises (5.20)

Use nested loops that display the following patterns in four separate programs:

Pattern 1:




```
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
1 2 3 4 5 6
```

Pattern 2:




```
1 2 3 4 5 6
1 2 3 4 5
1 2 3 4
1 2 3
1 2
1
```

Pattern 3:



```
1
2 1
3 2 1
4 3 2 1
5 4 3 2 1
6 5 4 3 2 1
```

Pattern 4:



```
1 2 3 4 5 6
1 2 3 4 5
1 2 3 4
1 2 3
1 2
1
```

Extra Exercises (Homework)

From the Textbook

- Programming Exercises:
 - 5.19
 - 5.21
 - 5.22
 - 5.24
 - 5.26
 - 5.29
 - 5.41

From MyProgrammingLab (<https://pearson.turingscraft.com>)

- 5.3
 - 51176
 - 51177
 - 51259
 - 51250
 - 51268
 - 51269
 - 51286

Upload Your Solutions



Upload your solutions of the lab activities to Blackboard.