

# Lab Assignment Week 07

*CSC/DSCI 1301 – Principles of CS/DS I*

*Week of February 24<sup>th</sup>, 2025*

## Introduction

Welcome to the seventh programming lab of CSC/DSCI 1301! Today we will be covering the following topics:

- Creating while loops
- Creating for loops
- Using range() function

## Lab policy reminders:

- Attendance is mandatory.
- Labs must be completed **individually**.
- TAs are here to help you. Ask them for help!
- Lab assignments are due at the end of each lab.

## Comments

The lab assignment requires the inclusion of comments to enhance code readability and understanding. Specifically, a block comment at the beginning of the Python file is required. Your block comment should include the following:

- The program name
- The author's name (your name)
- A description of the program's overall purpose

Additionally, inline comments should be used throughout the code to explain specific lines or sections that might be less obvious to someone reading the code. These inline comments can clarify complex calculations, explain the purpose of certain variables, or provide additional context for specific code blocks.

## Deliverables:

1. Python files for all 3 programs in the lab
2. Screenshots of program output for all 3 programs

If you have any questions, please do not hesitate to ask your TA!

## Program 1: password.py

Many user-created passwords are simple and easy to guess. Write a program that strengthens a simple password by replacing characters using the key below and appending "!" to the end of the input string.

- o becomes 0
- i becomes 1
- a becomes @
- e becomes 3
- A becomes 4
- B becomes 8
- s becomes \$

### Example Output

```
Please Enter Your Password: IHaveAStrongPassword
Your new strong password is: IH@v34Str0ngP@$$w0rd!
```

### Skills Covered

- Creating for loops
- Utilizing if-elif-else statements

### Deliverables

For this program, you will need to provide the Python file containing your code as well as a screenshot of the output of your program. Please name your files as follows:

- Python Files
  - lastname\_firstname\_filename.py
  - For example: **hawamdeh\_faris\_password.py**
- Screenshots
  - lastname\_firstname\_filename.png
  - For example: **hawamdeh\_faris\_password.png**

## Program 2: reverse.py

Write a program that takes in a line of text as input and outputs that line of text in reverse. The program should repeat, allowing the user to enter multiple inputs. Your program should terminate only when the user enters "Quit," "quit," or "q" as input.

Hints: Individual characters of a string can be accessed like elements in a list. The range function can create a countdown sequence if given a negative **step** parameter and the appropriate **start** and **stop** parameters. Build a new **reverse\_string** by appending characters from the original string in reverse.

### Example Output

```
Please Enter Your String: The quick brown fox jumps over the lazy dog
Reversed: god yzal eht revo spmuj xof nworb kciuq ehT
```

```
Please Enter Your String: quit
```

### Skills Covered

- Creating while loops
- Creating for loops
- Utilizing the range() function

### Deliverables

For this program, you will need to provide the Python file containing your code as well as a screenshot of the output of your program. Please name your files as follows:

- Python Files
  - lastname\_firstname\_filename.py
  - For example: **hawamdeh\_faris\_reverse.py**
- Screenshots
  - lastname\_firstname\_filename.png
  - For example: **hawamdeh\_faris\_reverse.png**

## Program 3: normalize.py

When analyzing data sets, such as data for human heights or human weights, a common step is adjusting the data to make them more convenient. This adjustment can be made by normalizing values between 0 and 1 or throwing away outliers.

For this program, you must adjust the input values by dividing all inputs by the largest value. The first input begins with an integer indicating the number of floating-point values that will follow.

Output each floating-point value with two digits after the decimal point, which can be achieved as follows: `print(f'{your_value:.2f}')`

### Example Output:

```
Please enter the number of floating-point values: 5
Please enter a floating-point value: 30.0
Please enter a floating-point value: 50.0
Please enter a floating-point value: 10.0
Please enter a floating-point value: 100.0
Please enter a floating-point value: 65.0
```

```
Normalized Floating-Point Values:
0.30
0.50
0.10
1.00
0.65
```

### Skills Covered

- Creating for loops
- Using range() function

### Deliverables

For this program, you will need to provide the Python file containing your code as well as a screenshot of the output of your program. Please name your files as follows:

- Python Files
  - lastname\_firstname\_filename.py
  - For example: **hawamdeh\_faris\_normalize.py**
- Screenshots
  - lastname\_firstname\_filename.png
  - For example: **hawamdeh\_faris\_normalize.png**