CS 3035, Fall 2022
In-Lab Exercise 14

Due: October 10, 2022 (11:59 PM midnight)

This lab has two parts. Please attempt both parts. However, submission is only required for Part 1. Part 2 will be used for small group discussion in the classroom at the end of the lab period.

Part 1: Writing a Python script to convert from Celsius to Fahrenheit

Write a Python script that converts from Celsius to Fahrenheit, printing a table showing conversions for integer values between a minimum and a maximum both specified by the user. For example, if the user enters minimum -40 and maximum 100, the output will show each integer Celsius value from -40 to 100 with the corresponding Fahrenheit value.

Use the input() function to get user input. This function will be called twice, once to get the minimum and once to get the maximum value from the user input. Note that the function input() for user input is used in the testFunction.py script discussed in class and available on the class git repository in cs3035/python/week7.

Use a separate function for conversion from Celsius to Fahrenheit. The conversion formula is f = c * 9.0/5.0 + 32. Most of the Fahrenheit values will not be integers; do not worry about the varying precision in the output.

Turn in your Python code and the screenshot of the output from a sample run of your program.

Part 2: Iteration in Python

In class today, we learned various methods for iteration in Python. In this part, we will examine some aspects of the material we covered today. Please use the Python interactive shell to conduct the following exercises:

1. Use the zip() function to visit two lists *of unequal lengths* that you create and print parallel items from each list as a new list. Look at slide #12 from today's class material for how to do this. What do you observe when you use two lists of unequal length as arguments to zip()?

2. In Lab 13, you created a dictionary. Is it possible to create the same dictionary using the zip function where the keys and values are initially in the form of two lists? Try doing this exercise to see if it works.

3. Examine the properties of an iterable created to iterate over a range vs. one created to iterate over a zip output.

   3.1. **Does the range() function support multiple iterators?**

   Use a range function to return a sequence of 10 elements. Create two iterables for the same range variable. Slide #16 from today's class shows how to create a range iterable (iter). Iterate through the sequence using both iterators sequentially. For example, if iter1 and iter2 are the two iterables, do next(iter1) followed by next(iter2) followed by next(iter3) and so on. What do you observe?

3.2. **Does the map() function support multiple iterators?**

Conduct an exercise similar to 3.1. What do you observe?