

## Temperature Conversion Program (Function Version)

This program allows a user to convert a range of values from Fahrenheit to Celsius, or Celsius to Fahrenheit, as presented in Unit 3. In this version, however, the program is designed with the use of functions. This program utilizes the following programming features.

► value-returning functions

► non-value-returning functions

Example execution of the program is given below:

```
This program will convert a range of temperatures
Enter (F) to convert Fahrenheit to Celsius
Enter (C) to convert Celsius to Fahrenheit

Enter selection: F
Enter starting temperature to convert: 65
Enter ending temperature to convert: 95

Degrees Degrees
Fahrenheit Celsius
65.0 18.3
66.0 18.9
67.0 19.4
68.0 20.0
69.0 20.6
70.0 21.1
71.0 21.7
72.0 22.2
73.0 22.8
74.0 23.3
75.0 23.9
76.0 24.4
77.0 25.0
78.0 25.6
79.0 26.1
80.0 26.7
81.0 27.2
82.0 27.8
83.0 28.3
84.0 28.9
85.0 29.4
86.0 30.0
87.0 30.6
88.0 31.1
89.0 31.7
90.0 32.2
91.0 32.8
92.0 33.3
93.0 33.9
94.0 34.4
95.0 35.0
```

**Task:** In IDLE, open a new project and save it as TempConvert\_yourLastName. Copy the code from the sample on the next page. Test and revise the program, as needed.

```

1 # Temperature Conversion Program (Celsius-Fahrenheit / Fahrenheit-Celsius)
2
3 def displayWelcome():
4
5     print('This program will convert a range of temperatures')
6     print('Enter (F) to convert Fahrenheit to Celsius')
7     print('Enter (C) to convert Celsius to Fahrenheit\n')
8
9 def getConvertTo():
10
11     which = input('Enter selection: ')
12     while which != 'F' and which != 'C':
13         which = input('Enter selection: ')
14
15     return which
16
17 def displayFahrenToCelsius(start, end):
18
19     print('\n Degrees', ' Degrees')
20     print('Fahrenheit', 'Celsius')
21
22     for temp in range(start, end + 1):
23         converted_temp = (temp - 32) * 5/9
24         print(' ', format(temp, '4.1f'), ' ', format(converted_temp, '4.1f'))
25
26 def displayCelsiusToFahren(start, end):
27
28     print('\n Degrees', ' Degrees')
29     print(' Celsius', 'Fahrenheit')
30
31     for temp in range(start, end + 1):
32         converted_temp = (9/5 * temp) + 32
33         print(' ', format(temp, '4.1f'), ' ', format(converted_temp, '4.1f'))
34
35 # ---- main
36
37 # Display program welcome
38 displayWelcome()
39
40 # Get which conversion from user
41 which = getConvertTo()
42
43 # Get range of temperatures to convert
44 temp_start = int(input('Enter starting temperature to convert: '))
45 temp_end = int(input('Enter ending temperature to convert: '))
46
47 # Display range of converted temperatures
48 if which == 'F':
49     displayFahrenToCelsius(temp_start, temp_end)
50 else:
51     displayCelsiusToFahren(temp_start, temp_end)

```

### Notes:

In lines 3–29 are defined functions `displayWelcome`, `getConvertTo`, `displayFahrenToCelsius`, and `displayCelsiusToFahren`. The functions are directly called from the main module of the program in lines 32–48.

On line 35, the non-value-returning function `displayWelcome` is called. Its job is to display information about the program to the user. It does not need to be passed any arguments since it performs the same output each time it is called. Next, on line 38, value-returning function `get-`

`ConvertTo` is called. This function also is not passed any arguments. It simply asks the user to

enter either 'F' or 'C' to indicate whether they want to convert from Fahrenheit to Celsius, or Celsius to Fahrenheit. The input value entered is returned as the function value.

The instructions on line 41–42 then prompt the user for the start and end range of temperatures to be converted. This task does not warrant the construction of a function since there are only two input instructions to accomplish this.

The final part of the program displays the converted range of temperatures. Two non-value-returning functions are defined for accomplishing this task—`displayFahrenToCelsius` and `displayCelsiusToFahren`. Each is passed two arguments, `temp_start` and `temp_end`, which indicate the range of temperature values to be converted.

What is left to look at is the implementation of each of the individual functions. The implementation of function `displayWelcome` (lines 3–6) is very straightforward. It simply contains three print instructions. Function `getConvertTo` (lines 8–13) contains a call to `input` followed by a while loop that performs input validation. The user is forced to enter either 'F' or 'C', and is continually prompted to re-enter as long as a value other than these two values is entered. When the loop terminates, variable `which` is returned by the return statement in line 13.

Function `displayFahrenToCelsius` (lines 15–21) and function `displayCelsius-ToFahren` (lines 23–29) are similar in design. Each contains two parameters - `start` and `end` (which are each passed actual arguments `temp_start` and `temp_end` in the main section of the program). Each first prints the appropriate column headings followed by a `for` statement that iterates variable `temp` over the requested temperature range. The conversion formula is different in each, however. Each has the same final print instruction to print out the original temperature and the converted temperature in each of the columns.

### **Extension:**

Modify the Temperature Conversion program, so that it allows the user to select temperature conversion to include degrees Kelvin, in addition to degrees Fahrenheit and degrees Celsius. Include input error checking for inappropriate temperature values.