**4. Iteration (Conditional loops)**

**Task 1:**

**Plan:**

1. User enters a temperature.
2. Validation is used to check if the temperature entered is an integer, if it is not the user is asked to enter the temperature again, and an error message is displayed.
3. Once the user has entered a valid temperature (an integer), the temperature is checked to see if it is in range -40 to 55. If it is out of range an error message is displayed, and the user is asked to enter the temperature again.
4. If the temperature entered by the user meets both conditions above. Is an integer, and is in range -40 to 55. The temperature is appended to a list.
5. The process repeats from step 1, for seven temperatures. Once 7 valid temperatures have been entered, the program will accurately calculate the average temperature and output to user.

**Pseudocode:**

Temp 🡸 []

For I in range(8):

While len(temp) < i:

Try:

User 🡸 input(“Enter Temperature”,i)

If user in range(-40, 55):

temp.append(user)

Else:

Print(“Please only enters temperatures in range -40 to 55!”)

Except:

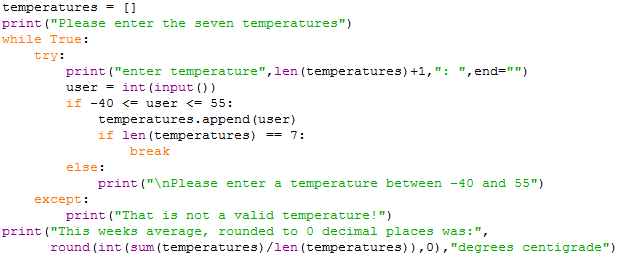
Print(“Incorrect Input!”)

Print(“Average Temperature was:”, round(sum(temp) / len(temp)))

**Variables table:**

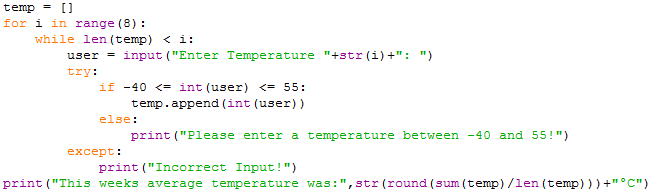
|  |  |  |
| --- | --- | --- |
| Variable Name | Data Type | Comment |
| Temp | List | Used to store the 7 valid temperatures. Also used in the process of calculating the average. |
| User | String | Stores the users input, when they enter a temperature. |

**Screenshot evidence:**

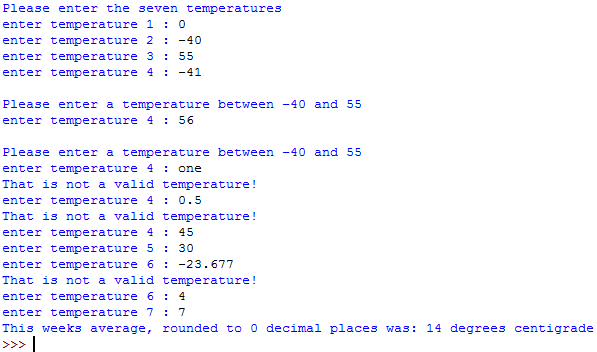
The screenshot bellow shows my first attempt at this task. In the screenshot a list variable “temperatures” is defined. Followed by a print statement instructing the user to “enter seven temperatures”. A while loop is then used for validation on each of the seven temperatures the user inputs, and a try and except statement is used to catch non-integer values. After each input if the try and except statement has said the value is valid. The value is then checked to see if it is in range -40 to 55. It is then the value is appended to the “temperatures” list, else an error message is displayed instructing the user to “Please enter a temperature between -40 and 55”. If the try and except statement rejects the value as being non-integer, then a different error message is displayed informing the user that their input was not a valid temperature. Once 7 valid temperatures have been entered, in the screenshot bellow a break statement is then used to forcefully break out of the while loop. Finally, the average temperature is accurately calculated and output.

The code above is not a very good example of how to approach this task, this is because the break statement show bad programming practise as the loop is not controlled. You could also just as easily have a for loop iterating 7 times, with a while loop inside of it validation the users input. There is no need for a break statement. Also in the print statement on the last line of the code above the average is not being calculated properly. For example, let’s say we sum the temperatures list and divide it by the length of its self, and get the float “9.714285714285714”. now in the screenshot above before we round the value the program is casting it to an integer, show by the “round(**int**(sum(temperatures)/ len(temperatures)))” calculation. This results in the float being cast to the integer “9” before being rounded, and well 9 rounded returns 9 as being your average. But this is incorrect because the float value was “9.7…”, so it should be rounded to 10. To fix this the float value must not be cast to an integer.

Below is my second attempt at this task; It shows accurate calculation of the average value, and has made use of controlled loops. It starts off similar to the solution above with the variable “temp” being defined as a list. A for loop then given the task of preforming 8 iterations is present, with a while loop for validation followed by a try and except statement used for checking if the users input is an integer. the while loop is linked to the for loop as it will only end when the length of the “temp” variable is greater than or equal to the value of I. this also results in the for loop having to loop 8 times rather than 7, because the while loop will only start running from the second iteration. Inside the while loop the user is asked to enter a temperature for value “I”, I is the counter variable that +1 is applied to after every iteration. The try and except statement is then used to check for non-integers, and an if statement used to check for out of range inputs. If the input from the user is valid (meaning it meets both conditions) then the value is appended to a “temp” list. after the for loop has passed through all its iterations and has finished, a print statement calculates accurately and outputs the average.

**Sample run (Output):**

Task 1 solution 1:



In the screenshot above I enter a valid input of 0. I then enter -40 and 55 to test that the range condition is working correctly, both values are excepted. Then just to make sure I enter -41, and 56, and on both inputs an error message is spat out telling me to “enter a temperature between -40 and 55”. I then enter a string value “one”, followed by a float 0.5. and again an error message is output saying “not a valid temperature!”. Finally, after I have entered 7 valid temperatures the average is output.

Task 1 solution 2:

**Flowcharts:**

START

Print(“Average Temperature was:”, round(sum(temp) / len(temp)))

Print(“Incorrect Input!”)

Except:

Print(“Please only enters temperatures in range -40 to 55!”)

No (Else)

temp.append(user)

Yes

If user in range (-40, 55):

User 🡸 input(“Enter Temperature”,i)

Try:

For I in range(8)

Temp 🡸 []