

Temperature Analyzer

Workshop 3 (worth 3% of your final grade)
URL: <https://.....>

In this workshop, you will code a C-language program that analyzes data logically.

LEARNING OUTCOMES

Upon successful completion of this workshop, you will have demonstrated the abilities:

- to create a simple interactive program
- to use a selection construct to code a decision
- to use an iteration construct to code repetitive logic
- to nest a logical block within another logical block
- to describe to your instructor what you have learned in completing this workshop

SUBMISSION POLICY

Your workshops are divided into two sections; [in_lab](#) and [at_home](#).

The "in-lab" section is to be completed during your assigned lab section. It is to be completed and submitted by the end of the workshop period. **If you attend the lab period and cannot complete the in-lab portion of the workshop during that period, ask your instructor for permission to complete the in-lab portion after the period.** If you do not attend the workshop, you can submit the "[in_lab](#)" section along with your "[at_home](#)" section, but there will be a reduction from 40% to 10% for the "[in_lab](#)" portion. The "[at_home](#)" portion of the lab is **due the day before your next scheduled workshop.**

All your work (all the files you create or modify) must contain your name, Seneca email and student number.

You are responsible for regularly backing up your work.

IN-LAB: ITEM CLASS (50%)

Download or clone workshop 3 from <https://....>

Code a program that does the following:

1- Before the declaration of main define NUMS as 3: > **#define NUMS 3** <

2- Print the title of the application.

> **==== IPC Temperature Analyzer ====** <

* the first character of the title must be on column 8

3- Using a for loop, prompt the user to enter the high and low values for each of NUMS days. The values entered must be between -40 and 40, and high must be greater than low.

Print the following messages:

> **Enter the high value for day 1:** < (or day 2, or day 3)

* Read the high value.

> **Enter the low value for day 1:** < (or day 2, or day 3)

* Read the low value.

4- Within the body of the for loop, use a while loop around the input statements to validate the input: high must be greater than low, high must be less than 41, and low must be greater than -41

*If any entry is incorrect, prompt the user to enter again until the entries pass the tests:

> **Incorrect values, temperatures must be in the range -40 to 40, high must be greater than low.** <

Then prompt again for the high and low temperatures for the day.

5- When the user has correctly entered the high and low temperatures, add them to variables that will store the total high and low temperatures for NUMS days.

6- When the for loop finishes calculate the average (mean) temperature for NUMS (the total of the temperatures entered divided by the number of temperature readings entered) days and display:

> **The Average (mean) temperature was: --** <

Output Example with Errors Handled

```
----- IPC Temperature Analyzer -----  
  
Enter the high value for day 1: 8  
Enter the low value for day 1: -2  
Enter the high value for day 2: 41  
Enter the low value for day 2: -4  
  
Incorrect values, temperatures must be in the range -40 to 40, high  
must be greater than low.  
  
Enter the high value for day 2: 9  
Enter the low value for day 2: -4  
Enter the high value for day 3: 5  
Enter the low value for day 3: 11  
  
Incorrect values, temperatures must be in the range -40 to 40, high  
must be greater than low.  
  
Enter the high value for day 3: 11  
Enter the low value for day 3: 5  
  
The average (mean) temperature was: 4.50
```

For submission instructions, see the [SUBMISSION](#) section below.

IN_LAB SUBMISSION:

To test and demonstrate execution of your program use the same data as the output example above, including the erroneous entries (the mistakes).

If not on matrix already, upload your [temps.c](#) to your matrix account. Compile and run your code and make sure everything works properly.

Then run the following script from your account: (replace profname.proflastname with your professors Seneca userid)

```
~profname.proflastname/submit 144_w3_lab <ENTER>
```

and follow the instructions.

AT_HOME: (40%)

After completing the [in_lab](#) section, upgrade [temps.c](#) to

- process a 4-day period using a single change to your in_lab code
- display the highest temperature, and on which day it occurred
- display the lowest temperature, and on which day it occurred
- calculate and display the mean temperature for the 4-day period.

Output Example

```
----- IPC Temperature Analyzer -----  
  
Enter the high value for day 1: 8  
Enter the low value for day 1: -2  
Enter the high value for day 2: 9  
Enter the low value for day 2: -4  
Enter the high value for day 3: 11  
Enter the low value for day 3: 5  
Enter the high value for day 4: 10  
Enter the low value for day 4: 3
```

The average (mean) temperature was: 5.00

The highest temperature was 11, on day 3

The lowest temperature was -4, on day 2

AT-HOME REFLECTION (10%)

Please provide brief answers to the following questions in a text file named `reflect.txt`.

- 1) Name all the iteration constructs?
- 2) Explain the difference between a “do while” and a “while” loop?
- 3) What is a conditional expression?

AT_HOME SUBMISSION:

The submission script is not ready yet, check back later.

~~To test and demonstrate execution of your program use the same data as the output example above... (8, 2, 9, 4, 11, 5, 10, 3)~~

~~If not on matrix already, upload your `temps.c` and `reflect.txt` to your matrix account. Compile and run your code and make sure everything works properly.~~

~~Then run the following script from your account: (replace profname.prolastname with your professors Seneca userid)~~

~~`~profname.prolastname/submit ipc_w3_at_home <ENTER>`~~

~~and follow the instructions.~~