

Logic

Workshop 3 (out of 10 marks - 4% of your final grade)

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

SUBMISSION POLICY

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 (with a penalty; see below). The "at_home" portion of the workshop is **due no later than four (4) days following the in-lab assigned date (even if that day is a holiday) by 11:59PM.**

Your work (all files you create or modify) must contain your:

- Full name
- Student number
- Seneca email address
- Section

You are responsible to back up your work regularly.

Late submission penalties:

- In-lab portion submitted late, with at-home portion: 0 for in-lab. Maximum of 70/70 for at-home and reflection
- If any of in-lab, at-home or reflection portions is missing the mark will be zero.

IN-LAB: (30%)

Download or clone workshop 3 (WS03) from <https://github.com/Seneca-144100/IPC-Workshops>

Code a program in `temps.c` 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 -----<
```

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.

IMPORTANT: You may only declare **two (2) int type variables** for the high and low values

4. Use a nested while (or do-while) loop to analyze the results, high must be greater than low, high must be less than 41, 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.

OUTPUT EXAMPLE WITH ERRORS HANDLED *(use this data for submission)*

----- 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

IN_LAB SUBMISSION:

To test and demonstrate execution of your program use the same data as the output example above.

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 and replace **SAA** with your section)

```
~profname.proflastname/submit 144w3/SAA_lab <ENTER>
```

and follow the instructions.

Please Note

- A successful submission does not guarantee full credit for this workshop.
- If the professor is not satisfied with your implementation, your professor may ask you to resubmit. Resubmissions will attract a penalty.

AT_HOME: (30%)

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

- 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 LOW temperature for the 4-day period
- Calculate and display the mean HIGH temperature for the 4-day period
- Calculate and display the mean OVERALL temperature for the 4-day period

Hints: In order to do the above, you will need to declare more variables.

- Accumulate the low and high temperatures independently
- Calculate averages after all the totals are known

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: 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
```

```
Enter the high value for day 4: 10
```

```
Enter the low value for day 4: 3
```

```
The average (mean) LOW temperature was: 0.50
```

```
The average (mean) HIGH temperature was: 9.50
```

```
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 (40%)

Please provide answers to the following in a text file named **reflect.txt**.

In three or more paragraphs, explain what you learned while doing this workshop. **Your answer should at least include the following:**

- How did pseudo-coding the solution beforehand help with the coding part? If you didn't use pseudo code (or prepare some kind of an outlined answer), think about your experience and either justify your method or explain why it may have been better to plan ahead.
- There are 3 types of iteration constructs. What ones did you use in your program, and why?
- What did you find interesting and/or difficult about this workshop?

Reflections will be graded on clarity of thought, correctness, grammar and spelling.

Note: when completing the workshop reflection it is a violation of academic policy to cut and paste content from the course notes or any other published source, or to copy the work of another student.

AT_HOME SUBMISSION:

To test and demonstrate execution of your program use the same data as the output example above... (8, -2, 9, -4, 5, 11, 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.proflastname with your professors Seneca userid and replace **SAA** with your section)

```
~profname.proflastname/submit 144w3/SAA_home <ENTER>
```

and follow the instructions.

Please Note

- A successful submission does not guarantee full credit for this workshop.
- If the professor is not satisfied with your implementation, your professor may ask you to resubmit. Resubmissions will attract a penalty.