This week has been easier for me than the previous week in which I couldn’t do the simplest of calculations. I learned a lesson from last week, so I started watching the tutorials before taking a look at the workshop pdf file. I first assured myself that I’m aware of the basics of if statement, looping, debugging, and the other materials taught; then I read the file and started coding.

I didn’t want to use pseudo-coding at all, to be honest. I thought that it is not going to be a hard problem to solve and I will be able to fix any problem that comes up during the coding process. After 30 minutes of struggling, I found myself stuck in a bunch of nonsense I have been writing; and the funny thing is that I blamed the compiler for not having the ability to figure out what’s going on in my code. I had to give up; so, I opened my notebook and explained the question in my native language and started to write what I thought would help me to calculate the right number. I mentioned every single detail the question asked for and then numbered the actions needed to be taken. The output was a clear step-by-step coding process, that could help me have a clear vision of what I want to do and stop me from losing track.

First thing first, we have to prompt the user to enter the highest and lowest temperature values for 4 days; we would also check the entered numbers so that the high value would be higher than the low value and high and low value would not exceed the +40 and -40, respectively. A for iteration is needed so that the actions mentioned would be done for 4 days and after that, it would exit the for loop and show the user the results. Inside the for loop, we have to put a while or do-while iteration in order to, first, analyze the numbers, and find the highest and lowest temperature in these 4 days. I didn’t see any benefit for applying a do-while; my pseudocode didn’t have a part so a thing would be executed first and then some conditions would be checked afterward. I tried to use a while construction to analyze the entered number but that just didn’t work out. Maybe it was because I’m still not that proficient in using a while loop to apply it properly. I used an if statement instead; my pseudocode said that if these statements are true (low > high, high > 40, …), prompt the user to enter the values again and correctly. For calculating the highest and lowest temp, a simple if statement would answer the question better than the other iteration constructs. I introduced an integer variable for the highest and lowest temp and gave them the initial value of 0. When the first for loop executes, the amount entered for the high and low value of day 1 will replace that 0 value. The important part starts from the second for loop, in which the if statement compares the values for day 2 and day 1, so if the high value for day 2 is higher than the one for day 1, it would replace it and would put it in the highest variable.

For the in-lab part of this workshop, although the question is simpler and easier to tackle, I came to this error that the average temperature was always shown as 0.00. I knew there was some problem with the variables I defined, but I couldn’t still fix it after a couple of attempts and redefining the variables as double or float. So, I put two breakpoints and started the debugger so that I would see the calculations being done in the background. I saw that the average is being calculated perfectly; the problem was when printf wanted to display that amount. After some research, I found that it would be fixed by adding a (double) or (float) in the printf statement just right before the totalTemp variable.