Learning Journal

The Learning Journal is a tool for self-reflection on the learning process. In addition to completing directed tasks, you should use the Learning Journal to document your activities, record problems you may have encountered and to draft answers for Discussion Forums and Assignments. The Learning Journal should be updated regularly (on a weekly basis), as the learning journals will be assessed by your instructor as part of your Final Grade.

Your learning journal entry must be a reflective statement that considers the following questions:

1. Describe what you did. This does not mean that you copy and paste from what you have posted or the assignments you have prepared. You need to describe what you did and how you did it.

2. Describe your reactions to what you did.

3. Describe any feedback you received or any specific interactions you had. Discuss how they were helpful.

4. Describe your feelings and attitudes.

5. Describe what you learned.

Another set of questions to consider in your learning journal statement include:

1. What surprised me or caused me to wonder?

2. What happened that felt particularly challenging? Why was it challenging to me?

3. What skills and knowledge do I recognize that I am gaining?

4. What am I realizing about myself as a learner?

5. In what ways am I able to apply the ideas and concepts gained to my own experience?

Finally, describe one important thing that you are thinking about in relation to the activity.

Your Learning Journal should be a minimum of 500 words.

This week (week 2), I learned about pseudocode and flowcharts. I learned about these through the resources and pdfs provided in the learning guide, as well as a series of youtube videos. They are both tools used to visualize the solution to a coding problem at a high level (ie without actually writing the code).

Pseudocode is using regular, understandable, language to describe each step in a program. It is much simpler and easier to understand than code of some other CS languages.

Flowcharts are .. flowcharts. I have used these before, but not the specific methods employed for computer code, such as different shapes representing processes, decisions, et al.

I have never had experience with using either of these tools in the context of programming before, and so, for the programming assignment, where we were given a choice to use one or the other, I decided to try both.

I found that I actually came up with slight different solutions. My flowchart did a separate check on the user’s input, to verify that it was 1, 2, 3, or 4, whereas in my pseudocode, I simply outlined IF and ELSEIF statements for each choice 1, 2, 3, and 4. Then if the user input anything else – they would be prompted. This might not be the explicit check that the assignment was looking for – but it guarantees the same result…

I found that pseudocode seemed to be much simpler, and more closely resembled what my final codes solution would look like. The flowchart still presented some junctions where I would need to make choices about what functions or python code to use. For eg. which type of function to evaluate each choice (1-4). The flowchart also seemed quite tedious – however I imagine it might be useful for very complex problems involving many steps – where you can break down the chart into different modules, and visually represent each decision. I suspect I may have found the pseudocode so simple because I was already able to formulate the solution to the problem with these conceptual tools – but that is not the case in which these conceptual tools are meant to be useful for.