**Reflection on workshop 4**

Name: Soham Thaker Student ID: 011-748-159 Email: sdthaker1@myseneca.ca

**What I learnt while doing this workshop?**

First, I learned how to store the user data into an array one by one. As the counter increased, it also changed the index of an array and this helped me store the values entered by the user regarding the high & low temperature into the respective arrays including their respective index.

I learned that for loop and arrays go together. The for loop makes the life of a programmer easier for storing the data into an array. I tried while and do while loop with arrays, but it was not comfortable in storing the data one index at a time. And even if I did that, I had to deal with other issues like data inconsistency and my other parts of the program were giving me errors and issues where I was reusing the same data.

Using an array is very convenient. For example, in this workshop, when I had to calculate the average of the high and low temperature along with overall average temperature of all the days, all I did was run a for loop and started assigning the values to another variable with the help of variable used in the for loop which corresponded to the index number of the array. Once that variable had all the values, I could do the computation like finding the average high, low, and overall temperatures very easily.

Furthermore, I learnt that iteration variables should be initialized with the value of 0, since the index of an array starts with 0 and this method also helps in storing the data to their respective indexes, along with certainty of the data.

Lastly, I learnt the importance of the macro definition “#define” or using constant integer variables, when dealing with arrays. This style of coding helps in alteration regarding assigning the size of an array. If I want to change the size of an array; I only need to do it at one place.

**Benefits of using arrays over any other way to store the data.**

Firstly, using an array into a program gives a programmer the flexibility to store the data, do computations and do almost anything with that data. One example is this workshop where students had to store the data into an array and use that data to compute the average temperatures. Another benefit is, using arrays in coding is very efficient way to deal with data and the compile time reduces dramatically when arrays are used compared to running a for loop and doing everything inside that loop, just as the students did in the last workshop.

Also, arrays provide a better solution where the data is independent. Once the data is stored into an array, programmers can easily access and change the data stored into the array. Programmers can access the values stored inside the array throughout the program if it is within the defined scope. If he or she try to use or change the data of that array outside the scope, then, it would be impossible to access the data stored into an array.

Furthermore, arrays are structured data or collections of same type of data. So, this helps in storing and accessing data as it stores the memory contiguously or in a linear way. So, to access that data stored into the memory, it makes easier and faster for the CPU to fetch that data and do the tasks as defined by the programmer.

Other way to deal with this workshop is to run a for loop and do everything, including the computation inside the for loop. For example, in last workshop, ws3, students had to run a for loop and everything including calculating overall mean, lowest day, highest day, average low and average high was done inside the for loop. However, this makes the program lengthy and it does not give the full freedom of computing other data among other functionalities. Also, there are compromises in using this method and the students would not be able to fully compute what is asked to do in this workshop. One example is, printing the average temperature up to the number of days entered by the user, as seen in this workshop. Here, if the students do not utilize array concept then, its very tough and almost impossible to find that specific data.

Also, if students use this way to work on this workshop, it is way too complicated if they are to take those values into functions because, if they were to take these values and pass them into a function, it would be almost impossible to do that. However, if they were to use arrays then, they can directly pass the array’s address into the function call as an argument and use a pointer or pass the original array as an argument into another array define within the function to receive that array’s contents and use the contents to do anything in the function itself.

**Why using parallel arrays is good?**

Parallel arrays are a type of arrays that hold different values at the same index. So, lets say a programmer has to deal with a set of data that go together and hand-in-hand; parallel arrays will make it easy to store the data and display that data or do any other operations on it. For example, mostly, temperatures have two values, high and low. Now, to find average of daily high and low temperature, a parallel array will hold daily temperature values into the two separate arrays as defined by the programmer. So if he or she runs a for loop, it would be very easy to find the daily average, since the data at the respective index is already available, all he or she has to do is find the average of that day by using a formula and store or print it out to the user right inside the for loop or send to another function or store that data to another variable. This is just one of the examples where parallel arrays can be helpful. In short, parallel arrays are helpful in certain situations, for example, where the data inside the arrays are inter-related and a programmer needs to perform further operations on that data.

**Why is it best practise to initialize iterator variables to values of zero?**

Firstly, by default, the index of an array starts with 0. So, the first index of an array is always 0. When declaring the iteration variables to 0, the iteration starts with the value of 0 itself and runs till “element – 1”, since elements are always 1 more than the last index. So, when iteration variables are initialized with a value of 0, there is no chance of assigning incorrect values to a particular index of an array as far as the condition keeps the iteration variable within the bounds of an array. As the iteration variable increases, it also assigns the correct value to the index number corresponding to the iteration variable. Also, it helps eradicate the complications and confusion regarding what value should be stored in that specific index of the array. Also, by using this approach, data can be handled properly according to the rule book.

In conclusion, an iteration variable should always be initialized with a value of 0, as far as the programmer is incrementing the variable. If he or she are decrementing the variable than the iteration variable must start with a value of “element - 1” and run until 0. There are many other logics to initialize and start the loop with different values but when a programmer is gradually incrementing a value of an iteration variable, the best practise is to initialize them with the value of 0.

**Conclusion**

In a nutshell, arrays are data structures which possess immense amount of potential to store a data. They are widely used among all the languages and it is a very deep concept to understand. Once a programmer knows how to work with arrays, he or she can handle immense amount of data easily. Along with its pros, there are cons as well just like a coin has two sides. But overall, arrays are an integral part of any programming language and if a person needs to be a successful programmer, knowing how to work with arrays is mandatory.