Critical Reflection

**Week 1 (Queues)**

**Task A**

For the first task I was asked to produce a application to store a queue of names and could take up to 10 names and could add, remove, display and show the number of names in the queue. This was a straightforward task which required basic programming skills using one class and an overloading constructor with a few methods. Within the main I used a switch and case statement to create a menu. The only thing I struggled with was getting my head around the idea of having the queue looped round to the start but once I used the Moodle resources it was straight forward.

**Task B (LinkedListGen)**

For the task B I was ask to create an application to manage a LinkedList of Library Books which was capable of adding, removing , displaying and sorting by ISBN. Whilst I managed to create all the methods with ease it took some trail and error to figure out the implementation of the book class. I soon figured out I would create a separate array of ISBN as this would be easier to sort and to be simpler. I then created a sorting method and used the “.sort” method using arrays to store the ISBN.

**Week 3**

**Task A (Binary Search Tree)**

for this task I was asked to create a binary search tree which inputs a text file and displays it as well as the amount of words in the tree and the height of the tree for this I used multiple methods to display including “preOrder”, “postOrder” and “inOrder”. Due to the understanding of the previous task I picked up the ideas and understandings of the task well using the materials on Moodle however the only part I struggled with is inserting the data into the graph using the read file method in the main as the code took some editing however after this I managed to get it to function as intended. Another issue I encountered was accessing the BSTree list within the main as both the static void main and the readfile method needed to access it however I overcame this by placing the initialisation of the BSTree outside the main in the class and declaring it as static.

**Task B (AVL tree)**

For this task it had the same methods as task A however with the added bonus of having balance factor this is where the tree is balanced correctly both left and right branches if it is within plus or minus 2 then it is balanced else the tree is rotated left or right and is swapped between each other. The most difficult part of this was the understanding behind however the actually coding part wasn’t too bad however I kept on having an issue with my visual studio which said that the certain parts of the code were missing when they really weren’t.

**Week 4**

**Task A (GenSelectSort)**

Task A part 1 asked me to calculate the big o notation this is simple as it was explained on Moodle and I simply assigned a value to each line and then calculated it by simplification (discarding constant ,clearing coefficients and picking most significant terms). For task A Part 2 I was asked to create a selection sort function which takes any data type array and sorts into the correct order. I took a array of books and sorted them by title as well as a integer array and string array. I found it was easier to create a separate class and call it via one line and used generic T type for the array so any thing could be passed.

**Task B (Greedy Algorithm)**

For this task I was asked to create a application which showed the highest combination of time slots which could be used for the computer lab by taking the start and finish times and creating a greedy algorithm which calculated the most amount of activates preformed. For this I created a simple algorithm. The hardest thing was trying to order the list however after some research and trial and error I found out how to create a sorted list by one of the objects inside of the list.

**Week 5**

**Task A (GraphNode)**

For this task I was asked to create graph which allowed to insert a node into the graph and insert a directed edge as well as display the total amount of nodes within the graph as well as the total number of edges. Due to my time off working I found it harder to get back into the rhythm of programming and took a while to understand the methods used and how they were implemented however once I had the hang of it I could understand what I was doing.

**Task B (Graph Traversals)**

For this task I was asked to create an airport connections by having different codes and connecting different airports then using the graph traversals to look up the connecting airport.as there was pseudocode already provided creating the code wasn’t to hard however I kept on a having a persistent error which caused my code not to work however after spending time on troubleshooting I found that when calling the variable current the assigned value which was calling from another method was null as it was never meeting the criteria due to being a different data type.

**Week 6**

**Task A(C++ pointers)**

For this Task we were asked to create a pointer arithmetic which pointed at a assigned value by the user within string array and then was replaced by a char value also assigned by the user and then a final result was displayed in the console with the replaced character. As this was a new language I was finding it hard to get used to the syntax which I found was much more low level when compared to C# and harder to understand as there was less functionality that was straight forward.

**Task B (Split)**

For this task we were asked to create a recombination using s1 and s2 that could be obtained using s3 this was by far the hardest due to only just starting C++ therefore I had to do research into this I found I would use the methods find, substr, erase and split. However even after doing this I don’t understand the task completely however I ran out of time therefore next time I would spend a lot more time on this and try to complete it with a higher standard.