Start of 10%

Date	Activities
15th January 2022	- I read the project instructions and every expectation of it - I figured the simple and difficult methods that I was going to implement. - I then outlined the harder method implementations and tried to figure out where the difficulty was
16th January 2022	- I revisited last semester's notes by quickly scanning through arrays on canvas and how to implement them, I realised I remember many things about arrays - I then had lunch with a friend who took data structures last semester and he made me realize some easy ways to implement some of the difficulty methods through our conversations - I then created two classes, the NumList class, that basically had skeleton methods. - After creating the skeleton methods, I then created my junit test class and considered most scenarios - I then created the NumArrayList class that initially had only one constructor and most of the methods were skeleton overridden methods except for the size() and capacity() because they were easy to implement.
18 January 2022	- I faced a challenge navigating Eclipse so I had to move my code to drJava where I am familiar with most of the feature - After re-reading the project's instructions I realised the NumArrayList class needed two constructors, one that takes an int value and another that does not take any parameter. - I implemented the instruction without any problem. - I continued to the add method that didn't take me long either
End of 10%	End of 10%

Start of 50%	Start of 50%
19th January 2022	- I created a Demo class that essentially had the main static method that helped me run my methods and saw what was going on with my implementations. - I implemented the toString method because I needed to see what my list was becoming after I ran the Demo class. - At first the toString method printed numbers in the list without any spaces, I then appended a space after every character to the StringBuilder I was using but there was an extra space after the last element of the list. - I added two if statements that specified to add a space after every element except for the last element of the list.
20th January 2022	Because I wasn't very confident with whatever I was doing, I went to the TA's office hours on zoom, I shared my screen, explained what I was doing and the TA assured me that I was doing the right thing.
21st January 2022	- I finished writing the rest of the methods and they were working perfect for the scenarios I tested with my demo, but they were not working quite well with some of my tester methods in my junit class. - The removeDuplicate method especially was not working well as it removed only duplicates that happened only 3 times, if a number was repeated 4 or more times, the method only removed 2 of them and left the rest - I also realised that my junit test class did not consider some scenarios, so I revisited it and added tests for different edges.
End of 50%	End of 50%
Start of 90%	Start of 90%

23rd January 2022	● I fixed the issue with removeDuplicate by creating a private helper method that checked for the list to see if there are duplicates. If duplicates are available in the list, the removeDuplicate is recursively called to remove the rest of the duplicates. ● I also realised that my list capacity was not expanding whenever it was filled up i.e size() == capacity(), I therefore created a pirate helper method, expand(), that expanded the list by increasing the capacity whenever the list was full. ● I then incorporated the expand() in the methods that needed to consider expanding the capacity such as insert and add
24th January 2022	● I created a main method inside my NumArrayList for demonstrations ● I debugged a few codes that were inconsistent with the demonstrations and the junit class
25th January 2022	● I revisited my code and added more comments to make the codes more readable ● I tested JavaC ● I was read to submit but needed to talk again to the TA to ensure that I had completed everything
End of 90%	End of 90%
Start of 100%	Start of 100%
27th January 2022	● I re-read the projects, instructions and the codes I had written. I shorten some necessary comments and cdes ● I finished writing the progress report and submitted