

CSCI 230 FALL 2023  
INSTRUCTOR: MICHAEL LEVET

---

## Homework Two: Revision

---

Oscar Jiang

9/22/2023

## 1 Homework Two: Bug Fixes

- Added size value and removed all instances of size(): No longer will the linked list be traversed to find the size. Instead, it will be set to zero by default for this program and will increment and decrement based off methods used.
- For contains method: Removed redundant condition to avoid errors- the search should stop and end completely when the node is null and will not check the null's element. Added the condition within the while loop, stopping the loop when needed. The contains method should not return a null error exception when the element isn't in the linked list but a false instead.
- For findFirstOccurence method: Removed the same redundant condition as contains method and fixed the same errors. Should return -1 instead of error.
- For remove methods: Fixed issue where head and tail were not set correctly after they were removed at their position. Before head and tail would be seen as any other node and therefore be removed, but not reset. Prevented errors(null exception error) this way too since head would try to set its previous node(null)'s next to this.head.next... same with this.tail .
- For removeAll method: traverses the linked list once. Removed the pointless calls to find and remove methods, optimizing the efficiency at which it removes.

## 2 Homework Two: Conceptual Difficulties and Lessons Learned

- For this assignment, I had great difficulty visualizing the nodes and their attributes' functions. Eventually, I understood it via drawing pictures etc. In addition, I did not know we had a size() method until after the project was nearly finished, allowing me to fix the insert method. Finally, the last difficulty I had was visualizing the traversal of the list.
- After completing the project, I can say in full confidence that I've grasped the concept of linked lists. However, there were revisions to be made. From those revisions, I realized how computationally heavy traversing the list was and therefore realized my mistake of doing so over and over using removeAll(old method) was. I also learned the conditions to stop a loop effectively with linked lists.