Midterm Test Skill	
Course Code: CPE201L	**Program:** Computer Engineering
Course Title: Data Structures and Algorithm	**Date Performed:** September 6, 2025
Section: CpE - 2A	**Date Submitted:** September 6, 2025
Name: Balaoro, Judge Wayne B.	**Instructor:** Engr. Maria Rizette Sayo

1. Objectives

1. Implement a singly-linked list of odd integers from 1 to 30 and do the following operations such as displaying all data, appending and deleting a node.
2. Save a copy to Github

2. Discussion

This activity is focused on implementing a singly linked list as I chose the 2nd option, it is a data structure where each node contains data and a pointer to the next node. Methods such as append, display and delete are involved in this chain by following the nextNode pointers from the head. My code demonstrates these core principles by managing nodes.

3. Materials and Equipment

Google Colab. Github

4. Procedure

Firstly, I created a class node and named it LinkedListNode . It has attributes such as odd and nextNode. I set the default of nextNode to None since when we append a Node that node will automatically point to None. Next, I also created a class LinkedList where all the methods reside and I put the attribute head and tail where I set the default to None. The first method I included inside the class LinkedList is the append to insert a Node. I put a conditional statement there that if the head is None, that node will become the head and tail as it is the first one and else if there's a head that newly added node will become the tail. I also set the pointer nextNode to link the data equivalent to 1st.next = 2nd etc.

The next method I added is display, I used a while loop to check if the linked list is empty it will stop and used an end separator so that it won't be printed into a new line for readability. The last method is delete, I used a conditional statement inside the while loop. I named a variable here as previous as it is necessary since the current node doesn't know who came before it, I have to keep track of the previous node to display the linked list. To also fix the tail problem since if I delete the tail, that tail will still be the tail that the system recognized even if i deleted it, so I added another conditional statement to check if the current node is the tail, the previous one will become the new tail. Lastly, I named LinkedList as ll for easy typing and appended each odd number from 1 to 30. I used the display method and after using the delete method for many situations, I used display method again to show the results of it.

5. Output

Implement a singly-linked list of odd integers from 1 to 30 and do the following operations: a.) Display all data b.) Append a Node c.) Delete a Node

```
class LinkedListNode:
    def __init__(self, odd, nextNode = None):
        self.odd = odd
        self.nextNode = nextNode

class LinkedList:
    def __init__(self, head = None, tail = None):
        self.head = head
        self.tail = tail

#method for b
    def append(self, odd):
        node  = LinkedListNode(odd)
        if self.head is None:
            self.head = node
            self.tail = node
        else:
            self.tail.nextNode = node
            self.tail = node

#method for a
    def display(self):
        current = self.head
        while current is not None:
            print(current.odd, "->", end = " ")
            current = current.nextNode
        print(None)

#method for c
    def delete(self, odd):
        current = self.head
        previous = None
        while current is not None:
            if current.odd == odd:
                if previous is not None:
                    previous.nextNode = current.nextNode
                else:
                    self.head = current.nextNode

                if current == self.tail:
                    self.tail = previous
                return
            previous = current
            current = current.nextNode
```

```
ll = LinkedList()
#b.) Append a node
ll.append(1)
ll.append(3)
ll.append(5)
ll.append(7)
ll.append(9)
ll.append(11)
ll.append(13)
ll.append(15)
ll.append(17)
ll.append(19)
ll.append(21)
ll.append(23)
ll.append(25)
ll.append(27)
ll.append(29)

#a.) Display all data
ll.display()


#c.) Delete a node
print("\nAfter Deleting a Node")
ll.delete(1)

ll.display()
```

```
1 -> 3 -> 5 -> 7 -> 9 -> 11 -> 13 -> 15 -> 17 -> 19 -> 21 -> 23 -> 25 -> 27 -> 29 -> None

After Deleting a Node
3 -> 5 -> 7 -> 9 -> 11 -> 13 -> 15 -> 17 -> 19 -> 21 -> 23 -> 25 -> 27 -> 29 -> None
```

6. Conclusion

In this midterm skill test activity, I was able to learn new things especially on the delete method. Since there are many conditionals like when you delete a head the next one should become the new head and if you delete a tail, the previous one should become the new tail. I was hesitant at first on naming it as previous since this is a singly linked list not doubly linked list but I thought that it is necessary for the delete method to work properly. Also, I considered using things such as 1st.next = 2nd, 2nd.next = 3rd but when I checked my code I already did this even if this doesn't look the same. I already linked my nodes on my append method. Overall, this skill test activity as our midterm helped me improve my logic and data structure knowledge with problem solving.

Lab Activity Rubric

Criteria	Ratings						Pts
SO 7 PI 1 **Student Outcome 7.1** Acquire and apply new knowledge from outside sources. threshold: 4.8 pts	6 pts Excellent | Educational interests and pursuits exist and flourish outside classroom requirements,knowledge and/or experiences are pursued independently and applies knowledge learned into practice	5 pts Good | Educational interests and pursuits exist and flourish outside classroom requirements,knowledge and/or experiences are pursued independently	4 pts Satisfactory | Look beyond classroom requirements, showing interest in pursuing knowledge independently	3 pts Unsatisfactory | Begins to look beyond classroom requirements, showing interest in pursuing knowledge independently	2 pts Poor | Relies on classroom instruction only	1 pts Very Poor | No initiative or interest in acquiring new knowledge	6 pts
SO 7 PI 2 **Student Outcome 7.2** Learn independently threshold: 4.8 pts	6 pts Excellent | Completes an assigned task independently and practices continuous improvement	5 pts Good | Completes an assigned task without supervision or guidance	4 pts Satisfactory | Requires minimal guidance to complete an assigned task	3 pts Unsatisfactory | Requires detailed or step-by-step instructions to complete a task	2 pts Poor | Shows little interest to complete a task independently	1 pts Very Poor | No interest to complete a task independently	6 pts
SO 7 PI 3 **Student Outcome 7.3** Critical thinking in the broadest context of technological change threshold: 4.8 pts	6 pts Excellent | Synthesizes and integrates information from a variety of sources; formulates a clear and precise perspective; draws appropriate conclusions	5 pts Good | Evaluate information from a variety of sources; formulates a clear and precise perspective.	4 pts Satisfactory | Analyze information from a variety of sources; formulates a clear and precise perspective.	3 pts Unsatisfactory | Apply the gathered information to formulate the problem	2 pts Poor | Gather and summarized the information from a variety of sources but failed to formulate the problem	1 pts Very Poor | Gather information from a variety of sources	6 pts
SO 7 PI 4 **Student Outcome 7.4** Creativity and adaptability to new and emerging technologies threshold: 4.8 pts	6 pts Excellent | Ideas are combined in original and creative ways in line with the new and emerging technology trends to solve a problem or address an issue.	5 pts Good | Ideas are creative and adapt the new knowledge to solve a problem or address an issue	4 pts Satisfactory | Ideas are creative in solving a problem, or address an issue	3 pts Unsatisfactory | Shows some creative ways to solve the problem	2 pts Poor | Shows initiative and attempt to develop creative ideas to solve the problem	1 pts Very Poor | Ideas are copied or restated from the sources consulted	6 pts

Total Points: 24