

MIDTERM PRACTICAL EXAM	
<b>Course Code:</b> CPE 201L	<b>Program:</b> BSCPE
<b>Course Title:</b> Data Structure and Algorithm	<b>Date Performed:</b> September 6, 2025
<b>Section:</b> 2A	<b>Date Submitted:</b> September 6, 2025
<b>Name:</b> Villanueva, Bryan O.	<b>Instructor:</b> Engr. Maria Rizette H. Sayo
<b>1.Objectives</b>	
<ul style="list-style-type: none"> <li>To implement a singly linked list of odd integers from 1 to 30.</li> <li>To apply operations such as displaying all the data, appending and deleting a node.</li> </ul>	
<b>2. Discussion</b>	
<p>A linked list is a linear data structure where each element called a node contains data and a reference to the next node. I created two classes, Node and LinkedList. The Node class holds the data and the pointer to the next node, while the LinkedList class manages the list itself. To use the three operations such as displaying the data, appending a node, and deleting a node. You will need to define each one of them. The display will make it possible to show all the odd integers in the linkedlist, while the append will make it possible for the user to add a new integer in the list, and delete to make it possible for the user to remove an integer in the list.</p>	
<b>3. Materials and Equipment</b>	
<ul style="list-style-type: none"> <li>Google Colab – A compiler for python</li> <li>PC – Equipment use in programming</li> <li>Github – For organizing and submitting of source codes and files</li> <li>Window Operating System</li> </ul>	
<b>4. Procedure</b>	
<ol style="list-style-type: none"> <li>1. Define a Node class with data and next attributes.</li> <li>2. Created a Linkedlist class with methods to display, append and delete data from the list.</li> <li>3. Make a list of odd integers from 1-30.</li> <li>4. Call the append method to print the updated linked list with added odd integer in the list.</li> <li>5. Call the delete method to print the updated linked list with deleted integer.</li> </ol>	

**Source Code:**

```
class Node:

    def __init__(self, data):

        self.data = data

        self.next = None

class SinglyLinkedList:

    def __init__(self):

        self.head = None

    def display(self):

        current = self.head

        if not current:

            return

        print("Linked List:", end=" ")

        while current:

            print(current.data, end=" -> ")

            current = current.next

        print("Null")

    def append(self, data):

        new_node = Node(data)

        if not self.head:

            self.head = new_node

            return

        current = self.head

        while current.next:

            current = current.next
```

```
        current.next = new_node

def delete(self, data):

    current = self.head

    if not current:

        return

    if current.data == data:

        self.head = current.next

        return

    prev = None

    while current and current.data != data:

        prev = current

        current = current.next

    if not current:

        return

    prev.next = current.next

linked_list = SinglyLinkedList()

for i in range(1, 31, 2):

    linked_list.append(i)

linked_list.display()

linked_list.append(33)

print("\nAfter appending 33:")

linked_list.display()

linked_list.delete(7)

print("\nAfter deleting 7:")

linked_list.display()
```

