Midterm - Skill Test	
Course Code: CPE 201L	**Program: BS Computer Engineering**
Course Title: Data Structures and Algorithms	**Date Performed:9/6/2025**
Section: 2A	**Date Submitted:9/6/2025**
Name: Catahan, Joshua A.	**Instructor:** Engr. Maria Rizette H. Sayo

1.Objectives

Choose only one (1):

1. Implement an array of even integers less than 50 but not less than 20 and do the following operations:
   a) Display the elements
   b) Find the maximum element
   c) Reverse the array
2. 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

2. Discussion

Data structures are fundamentals of any programming language. This is where we define the data type, data elements, and operations of a program. In data structures, there are different levels of organizing data such as arrays, linked-list, stacks, and queues. These structures have their own varying ways of accessing and storing data that each have their own right uses for a specific problem/situation. For instance, Linked-List is a type of data structure where each data element (node) contains an address/pointer to the next node. This makes it possible for insertion and deletion at all places of the linear list. Another example is an array; it is a type of data structure where the data elements are arranged in an orderly manner.

In this skill test for midterm, I chose number 2.

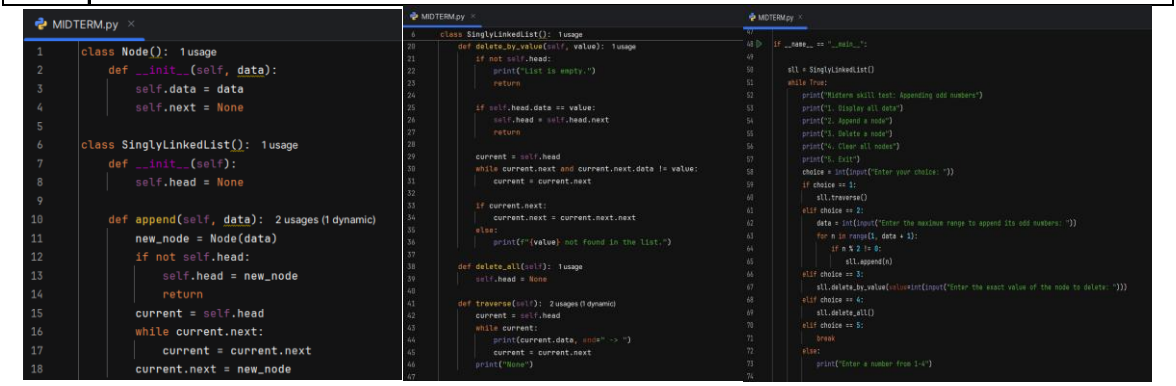
3. Materials and Equipment

- Github – Location where the file is uploaded
- Google Colab – Used to transfer file in github
- Python – Programming language
- Pycharm – Compiler used for coding

4. Procedure

1. I designed a node class to store data and the reference to the next code.
2. I constructed a LinkedLIst class with a head pointer initialized to None.
3. I implemented methods:
   - append() to add the odd numbers of the given range
   - delete_all() to delete all nodes
   - delete_by_value() to delete a node
   - traverse() to display all nodes.
4. I developed a menu-driven program with options to traverse, append nodes with the given range, delete all nodes and by value, or exit.
5. I incorporated input validation to handle invalid choices.
6. I executed and tested the program to verify correct functionality of all the methods.

5. Output



```
MIDTERM.py
class Node():  1 usage
    def __init__(self, data):
        self.data = data
        self.next = None

class SinglyLinkedList():  1 usage
    def __init__(self):
        self.head = None

    def append(self, data):  2 usages (1 dynamic)
        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
```

```
MIDTERM.py
class SinglyLinkedList():  1 usage
    def delete_by_value(self, value):  1 usage
        if not self.head:
            print("List is empty.")
            return

        if self.head.data == value:
            self.head = self.head.next
            return

        current = self.head
        while current.next and current.next.data != value:
            current = current.next

        if current.next:
            current.next = current.next.next
        else:
            print(f"{value} not found in the list.")

    def delete_all(self):  1 usage
        self.head = None

    def traverse(self):  2 usages (1 dynamic)
        current = self.head
        while current:
            print(current.data, end=" -> ")
            current = current.next
        print("None")
```

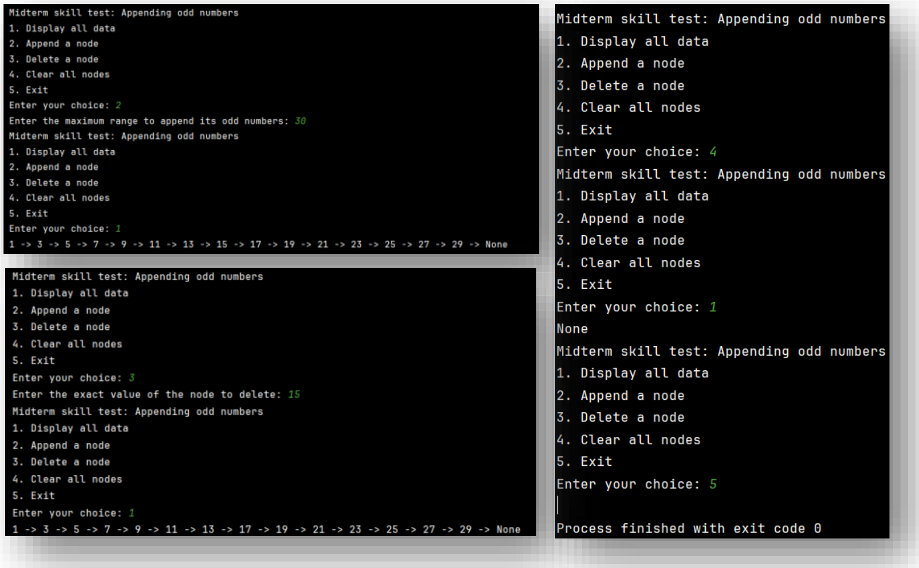
```
MIDTERM.py
if __name__ == "__main__":

    sll = SinglyLinkedList()
    while True:
        print("Midterm skill test: Appending odd numbers")
        print("1. Display all data")
        print("2. Append a node")
        print("3. Delete a node")
        print("4. Clear all nodes")
        print("5. Exit")
        choice = int(input("Enter your choice: "))
        if choice == 1:
            sll.traverse()
        elif choice == 2:
            data = int(input("Enter the maximum range to append its odd numbers: "))
            for n in range(1, data + 1):
                if n % 2 != 0:
                    sll.append(n)
        elif choice == 3:
            sll.delete_by_value(value=int(input("Enter the exact value of the node to delete: ")))
        elif choice == 4:
            sll.delete_all()
        elif choice == 5:
            break
        else:
            print("Enter a number from 1-4")
```

Figure 1: Screenshot of the program

In figure 1, it shows the code of the program. As seen above, this program was made by creating a node and linked list class. Inside these classes, its properties and methods (behavior) are defined.

Below are the process of how I tested this program, hence, completing the main objective of this test.

TESTING THE CODE:



```
Midterm skill test: Appending odd numbers
1. Display all data
2. Append a node
3. Delete a node
4. Clear all nodes
5. Exit
Enter your choice: 2
Enter the maximum range to append its odd numbers: 30
Midterm skill test: Appending odd numbers
1. Display all data
2. Append a node
3. Delete a node
4. Clear all nodes
5. Exit
Enter your choice: 1
1 -> 3 -> 5 -> 7 -> 9 -> 11 -> 13 -> 15 -> 17 -> 19 -> 21 -> 23 -> 25 -> 27 -> 29 -> None
```

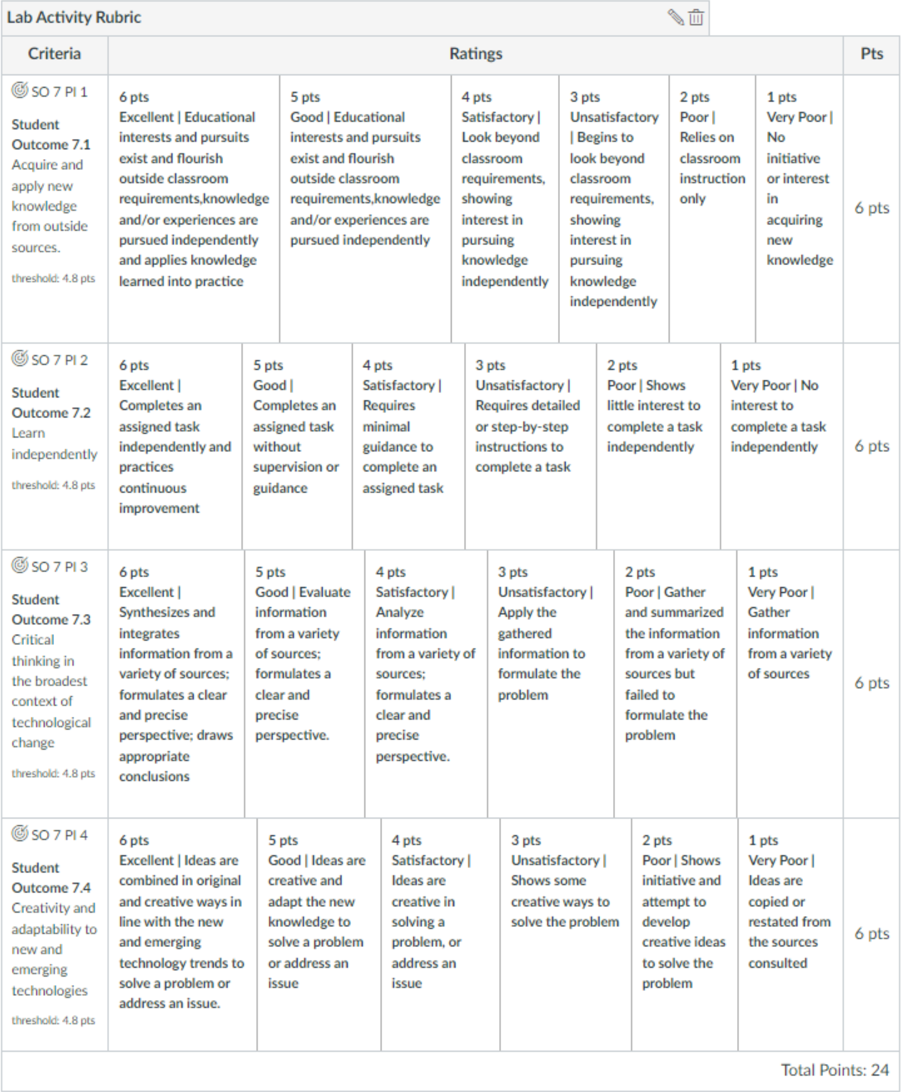
```
Midterm skill test: Appending odd numbers
1. Display all data
2. Append a node
3. Delete a node
4. Clear all nodes
5. Exit
Enter your choice: 3
Enter the exact value of the node to delete: 15
Midterm skill test: Appending odd numbers
1. Display all data
2. Append a node
3. Delete a node
4. Clear all nodes
5. Exit
Enter your choice: 1
1 -> 3 -> 5 -> 7 -> 9 -> 11 -> 13 -> 17 -> 19 -> 21 -> 23 -> 25 -> 27 -> 29 -> None
```

```
Midterm skill test: Appending odd numbers
1. Display all data
2. Append a node
3. Delete a node
4. Clear all nodes
5. Exit
Enter your choice: 4
Midterm skill test: Appending odd numbers
1. Display all data
2. Append a node
3. Delete a node
4. Clear all nodes
5. Exit
Enter your choice: 1
None
Midterm skill test: Appending odd numbers
1. Display all data
2. Append a node
3. Delete a node
4. Clear all nodes
5. Exit
Enter your choice: 5


Process finished with exit code 0
```

Figure 2: Screenshot of the program output

6. Conclusion

In conclusion, this midterm skill test was a challenging exam since it has taken me so much time in order to finish the program. One of the challenging parts in the objective was implementing a method to delete a node with a specified value. On the contrary, it was really easy implementing a method to delete all nodes. Aside from that, I think the easiest part of this program was implementing a way so that only the odd numbers are added in the singly list with a given range of 30. I did it through modulo (%), in which I remembered that it gives the remainder when dividing a number. Since I already know how to find the odd numbers, I just implemented a for loop so it will append it to the singly linked list. Overall this midterm skill helped me understand the level of my programming language, giving me the urge to improve more and excel in my chosen field.



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