

Midterm – Test Skill	
Course Code: CPE201L	Program: BSCpE
Course Title: Data Structures and Algorithms	Date Performed: 09 – 06 – 25
Section: 2A	Date Submitted: 09 – 06 – 25
Name: Regondola, Jezreel P.	Instructor: Engr. Maria Rizette H. Sayo
1.Objectives	
<ul style="list-style-type: none"> To implement a singly linked list in Python. To perform basic operations such as display, append, and delete in a linked list. 	
2. Discussion	
<p>In this activity, I created a program with two classes: Node and LinkedList. The Node class stores the data and the link to the next node, while the LinkedList class has the functions for display, append, and delete. In the main part of the code, I added all odd numbers from 1 to 29 into the list. I also made a menu with options so the user can choose to show the list, add a new node, delete a node, or exit.</p>	
3. Materials and Equipment	
<ol style="list-style-type: none"> Computer – Used to run and test the python program Python Interpreter (Google Colab) – Used to write the program and debug errors 	
4. Procedure	
<ol style="list-style-type: none"> Open Python or Google Colab Define the Node class with attributes for data and next. Define the LinkedList class with methods for display, append, and delete. Initialize the linked list with odd integers from 1 to 30. Implement a function to validate user menu choices. Create a looped menu system to allow the user to perform operations. Test the program by displaying, appending, and deleting nodes. 	
5. Output	
<p>Source Code:</p> <pre> class Node: def __init__(self, data): self.data = data self.next = None class LinkedList: def __init__(self): self.head = None </pre>	
<p>Figure 1. Defining the Node and LinkedList class</p>	

```

def display(self):
    current = self.head
    print("\nList of odd integers from 1 to 30:")
    while current:
        print(current.data, end=" -> ")
        current = current.next
    print("None\n")

```

Figure 2. Method for displaying the list

```

def append(self, value):
    new_node = Node(value)
    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, value):
    current = self.head
    prev = None
    while current:
        if current.data == value:
            if prev is None:
                self.head = current.next
            else:
                prev.next = current.next
            print(f"Deleted {value}.\n")
            return
        prev = current
        current = current.next
    print(f"Item {value} not found in the list.\n")
    prev.next = current.next

```

Figure 3. Methods to append and delete

```

def get_valid_choice():
    valid_choices = ["A", "B", "C", "X"]
    while True:
        choice = input(
            "Choose an option:\n"
            "A. Display all data\n"
            "B. Append a node\n"
            "C. Delete a node\n"
            "X. Exit\n"
            "Enter choice: "
        ).upper()

        if choice in valid_choices:
            return choice
        else:
            print("Invalid choice. Please try again.\n")

```

Figure 4. Function to get a valid choice from the user

```

if __name__ == "__main__":

    odd_list = LinkedList()
    for num in range(1, 30):
        if num % 2 != 0:
            odd_list.append(num)
        else:
            continue

    unfinished = True
    while(unfinished):
        user_input = get_valid_choice()

        if user_input == 'A':
            odd_list.display()
        elif user_input == 'B':
            to_append = int(input("\nEnter an item to append: "))
            odd_list.append(to_append)
            print()
        elif user_input == 'C':
            to_delete = int(input("\nEnter an item to delete: "))
            odd_list.delete(to_delete)
        else:
            unfinished = False

```

Figure 5. Main execution block

Output:

```
Choose an option:
A. Display all data
B. Append a node
C. Delete a node
X. Exit
Enter choice: A

List of odd integers from 1 to 30:
1 -> 3 -> 5 -> 7 -> 9 -> 11 -> 13 -> 15 -> 17 -> 19 -> 21 -> 23 -> 25 -> 27 -> 29 -> None
```

Figure 6. Display all data

```
Choose an option:
A. Display all data
B. Append a node
C. Delete a node
X. Exit
Enter choice: B

Enter an item to append: 31

Choose an option:
A. Display all data
B. Append a node
C. Delete a node
X. Exit
Enter choice: A

List of odd integers from 1 to 30:
1 -> 3 -> 5 -> 7 -> 9 -> 11 -> 13 -> 15 -> 17 -> 19 -> 21 -> 23 -> 25 -> 27 -> 29 -> 31 -> None
```

Figure 7. Append a node

```
Choose an option:
A. Display all data
B. Append a node
C. Delete a node
X. Exit
Enter choice: C

Enter an item to delete: 31
Deleted 31.

Choose an option:
A. Display all data
B. Append a node
C. Delete a node
X. Exit
Enter choice: A

List of odd integers from 1 to 30:
1 -> 3 -> 5 -> 7 -> 9 -> 11 -> 13 -> 15 -> 17 -> 19 -> 21 -> 23 -> 25 -> 27 -> 29 -> None
```

Figure 8. Delete a node

6. Conclusion

Through this activity, I was able to implement a singly linked list in Python and do some basic operations. I successfully displayed, appended, and deleted nodes in the list, which helped me understand how these functions work in a program. This activity also gave me more practice in applying what I learned to actual coding.

7. Rubrics

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								