|  |  |
| --- | --- |
| **Midterm Exam : Skill Test** | |
| **Course Code:** CPE- 201L | **Program:** BS-CPE |
| **Course Title:** Data Structure and Algorithm | **Date Performed:** September 06, 2025 |
| **Section:** 2-B | **Date Submitted:** September 06, 2025 |
| **Name:** Ljay L. Calica | **Instructor:** Ma’am Maria Rizette H. Sayo |
| 1. **Objectives** | |
| * To implement a singly linked list using Python. * To perform basic operations such as display, append, and reverse the list. | |
| **2. Discussion** | |
| * A linked list is a dynamic data structure where each node contains data and a reference to the next node, allowing for efficient insertions and deletions. This activity demonstrates how to implement a singly linked list and perform basic operations like appending nodes, displaying data, and reversing the list by manipulating node pointers. | |
| **3. Materials and Equipment** | |
| * Python Programming Language * Python was used to implement the singly linked list and perform the required operations. * Integrated Development Environment (IDE) * Google Colab | |
| **4. Procedure** | |
| 1. Define a Node class with two attributes, data to store the integer value and next to point to the next node in the list. 2. Create a LinkedList class with methods: append, display, and reverse. 3. Append numbers 1 to 30 to the list using the append() method. 4. Display the initial list, to ensure the numbers 1-30 is added. 5. Append number 31 and display the list again to confirm that the new node has been added. 6. Reverse the linked list and display the reversed order of nodes. 7. End procedure. | |
| **5. Output** | |
|  | |
| **6. Conclusion** | |
| * The activity demonstrated how a singly linked list works in Python, covering key operations like appending nodes, displaying the list, and reversing the order of nodes. Through this exercise, we gained a practical understanding of pointer manipulation and dynamic memory allocation in linked lists. | |
|  | |