|  |  |
| --- | --- |
| **MidTerm Skill Test** | |
| **Course Code: 201-L** | **Program: CPE** |
| **Course Title: DATA STRUCTURE** | **Date Performed: 06/09/25** |
| **Section: BSCPE 2-B** | **Date Submitted: 06/09/25** |
| **Name: Caasi, Karl Benedict D.** | **Instructor:** **Engr Maria Rizette Sayo** |
| 1. **Objectives** | |
| Append a new node at the end.  Reverse the entire list. | |
| **2. Discussion** | |
| A linked list is a linear data structure where each element called a node has two parts: the data and a reference or pointer to the next node. Unlike arrays, linked lists do not need a fixed size, so we can easily add or remove elements without shifting data.  . | |
| **3. Materials and Equipment** | |
| Google Colab Computer | |
| **4. Procedure** | |
| Create a Linked list class with the following methods:  append(data) → to add new nodes.  display() → to show all nodes.  reverse() → to reverse the list.  Insert integers 1 to 30 using the append() method.  Display the list.  Append a new node with value 30.  Display the updated list.  Reverse the entire list.  Display the reversed list. | |
| **5. Output** | |
|  | |
| **6. Conclusion** We were able to implement a singly linked list and perform display, append, and reverse operations. This activity helped us understand how linked lists work in memory and why they are more flexible than arrays when it comes to inserting or deleting elements. It also gave us practice in handling pointers (or references in Python), which are essential in data structures. | |
|  | |