|  |  |
| --- | --- |
| **SKILL TEST** | |
| **Course Code:** CPE 201L | **Program:** BSCPE |
| **Course Title:** Data Structure and Algorithm | **Date Performed:** 08/30/25 |
| **Section:** 2A | **Date Submitted:** 08/30/25 |
| **Name:** Villanueva, Bryan O. | **Instructor:** Engr. Maria Rizette H. Sayo |
| 1. **Objectives** | |
| * To understand and implement the concept of singly linked lists in Python. * To demonstrate how data can be stored and traversed using nodes and pointers. | |
| **2. Discussion** | |
| A linked list is a linear data structure where each element called a node contains data and a reference to the next node. Unlike arrays, linked lists don’t require contiguous memory, which makes them more flexible for dynamic data storage. 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. I used a string my full name as the data source and appended each character to the list. Then I traversed the list to print each character, showing how the data flows through the nodes. | |
| **3. Materials and Equipment** | |
| * Desktop Computer with Google Colab * Window Operating System | |
| **4. Procedure** | |
| 1. Defined a Node class with data and next attributes. 2. Created a LinkedList class with methods to append data and traverse the list. 3. Stored my full name in a variable called name. 4. Used a for loop to iterate through each character in the name and appended it to the linked list. 5. Call the traverse method to print each character from the list.   **Source Code:** | |
| **5. Output** | |
| Screenshot of your output based on the procedures. | |
| **6. Conclusion** | |
| This activity helped me understand how linked lists work and why they are useful in coding. I saw how each letter in my name could be saved in its own node and connected to the next one. Writing the code by myself made it easier to learn how pointers work and how to organize data step by step. I also learned that linked lists are good when you don’t know how much data you’ll have, or when the size changes often. Seeing how each part connects helped me understand how data moves in a program. | |
|  | |