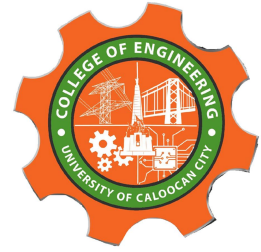




UNIVERSITY OF CALOOCAN CITY  
COMPUTER ENGINEERING DEPARTMENT



Data Structure and Algorithm

Laboratory Seat Work

---

# Python Data Structure

---

*Submitted by:*  
Enverzo, Kyle Andrey D.

*Instructor:*  
Engr. Maria Rizette H. Sayo

August-16-2025

## 1. What is a singly linked list, and how does it differ from an array?

A singly linked list is a linear data structure where each element (called a node) contains data and a pointer to the next node in the sequence. Unlike arrays, which store elements in contiguous memory locations, linked lists store elements in separate memory blocks connected via pointers. This allows dynamic memory allocation, whereas arrays have a fixed size once created.

## 2. When would you prefer a linked list over an array, and vice versa?

A linked list is preferred over an array when a program requires frequent insertions and deletions, especially in the middle of a sequence, because these operations can be performed efficiently without shifting elements. Linked lists are also useful when the size of the data set is not known in advance, as they allow dynamic memory allocation. On the other hand, arrays are preferred when quick, constant-time access to elements is needed through indexing, since accessing an element in a linked list requires sequential traversal from the head node. Arrays are also more memory-efficient in scenarios where the data size is fixed and predictable.

## 3. How are linked lists used in real-world applications?

**Browser history** – Implemented as a linked list where each visited page is a node, and moving back/forward navigates through the list.

**Undo functionality in text editors** – Each change can be stored as a node in a linked list, allowing step-by-step reversal.

**Music playlists** – Songs are linked, and you can move forward or backward easily.

## References

- [1] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, 3rd ed. Cambridge, MA: MIT Press, 2009.
- [2] M. A. Weiss, *Data Structures and Algorithm Analysis in C++*, 4th ed. Boston, MA: Pearson, 2013.
- [3] G. Brassard and P. Bratley, *Fundamentals of Algorithmics*. Englewood Cliffs, NJ: Prentice-Hall, 1996.