



**National University of Computer & Emerging Sciences, Karachi**  
**School of Computing (BSCS)**  
**Quiz 01 (Fall 2024)**



**Date: 05<sup>th</sup> September, 24 Time: 12:00pm –1:00 pm**

<b>Course Code: CS2001</b>	<b>Course Name: Data Structures &amp; Algorithms</b>
<b>Instructor Name: Ms. Ayesha Ali</b>	
<b>Student Roll No:</b>	<b>Section No:</b>

**MCQS (5 MARKS)**

- Which of the following is a benefit of using Abstract Data Types?
  - Reduces the complexity of algorithms.
  - Ensures that data is processed sequentially.
  - Allows for flexibility in implementing data structures.
  - Increases the memory requirements.
- Which of the following is a disadvantage of using a dynamic array?
  - Access time is slow.
  - It requires contiguous memory allocation.
  - It cannot be resized.
  - It uses more memory than necessary at all times.
- What happens when you try to access the next pointer of the last node in a singly linked list?
  - It points to the first node.
  - It causes an error.
  - It points to NULL or None.
  - It points to the previous node.
- Which of the following is NOT an advantage of a doubly linked list over a singly linked list?
  - Easier deletion of nodes without needing to traverse the list.
  - Easier traversal in both directions.
  - Requires less memory per node.
  - More flexible node insertion.
- What is NULL in the context of pointers?
  - A pointer to a memory address of zero.
  - A pointer that has not been initialized.
  - A pointer that points to a predefined memory location.
  - A constant value that represents an invalid pointer.

### SHORT Q/A (5 MARKS, 2.5 EACH)

#### Question 01

What is the advantage of using a singly linked list over an array?

#### Question 02

Explain how do you delete a node in a doubly linked list at position n?

### CODING (5 MARKS, 2.5 EACH)

#### Question 01

Write the code of Q2 above.

#### Question 02

Identify the error in the following code for inserting a node at the nth position in a singly linked list and provide a corrected solution.

```
void insertAtPosition(Node*& head, int data, int position) {
    Node* newNode = new Node();
    newNode->data = data;
    newNode->next = NULL;

    if (position == 0) {
        newNode->next = head;
    } else {
        Node* temp = head;
        for (int i = 0; i < position - 1 && temp != NULL; i++) {
            temp = temp->next;
        }
        newNode->next = temp->next;
        temp->next = newNode;
    }
}
```