1st Assignment

Data Structure and Algorithm (Java)

Deadline: 6th April 2025 **Marks: 15**

Important Instructions:

No late submissions allowed.

Submission through email is not permitted in any case.

Plagiarism will not be tolerated and will be graded as **ZERO.**

Make a Word document, add the executable against each problem statement, and submit the document file on the portal within the deadline.

Scenario 1:

You are building an order management system for an e-commerce website. The system should:

- Store customer orders using an ArrayList.
- Process orders in the order they were placed (FIFO).
- Remove an order once it is delivered.
- Retrieve the most recent order details.

Task:

- Implement the system using both ArrayList and LinkedList.
- Analyze the time complexity for adding and removing orders.
- Compare which implementation is better for an e-commerce system.

Scenario 2:

A stock market application needs to track stock prices dynamically. The system should:

- Add new stock prices as they arrive.
- Remove outdated stock prices.
- Retrieve the highest and lowest prices efficiently.

Task:

- Implement the system using both ArrayList and LinkedList.
- Compare the time complexity of insertion, deletion, and searching for min/max.
- Would a **Sorted Data Structure** improve performance?

Scenario 3:

A train booking system allows passengers to book seats. You must:

- Store the booking details using an **array** (fixed seat numbers).
- Allow dynamic allocation of seats using an **ArrayList**.
- Search for available seats efficiently.
- Cancel a booking and shift the remaining bookings accordingly.

Task:

- Implement the system using both **Arrays and ArrayLists**.
- Compare the time complexity of **booking, canceling, and searching**.
- Which approach is better for a large-scale system?

Scenario 4:

You are developing a comment section for a social media app where users can:

- Add comments in chronological order.
- Edit or delete any comment.
- Search for a comment by keyword.
- Display comments in reverse order (most recent first).

Task:

- Implement this system using **LinkedList**.
- Analyze the time complexity of insertion, deletion, and searching.
- Would an **ArrayList** be a better choice? Why or why not?

----------End of Assignment! ----------