



Department: CSE

Session: 2025-26

Programme: BTech, BCA

Semester: 3rd

Course Code: ENCS205, ENCA 201

Course: Data Structures

Assignment Details:

Theme: Foundation of Data Structures

Assignment Number: 01

Total Marks: 5

Contribution: 16% of internal evaluation

CO Mapping: CO1

Submission Requirements:

- Submit individually via GitHub and provide the link of submission by **5th September 2025**.
- Late submissions, copied assignments, or AI-generated text will not be accepted.
- Use the dedicated assignment format shared in classes.

Problem: Develop the weather data storage system

Description: A Weather Data Storage System is a software solution designed to systematically collect, store, and manage weather-related data such as temperature, humidity, and atmospheric conditions. In the context of this assignment, it focuses on organizing temperature records based on date and city using a structured data format like 2D arrays and Abstract Data Types (ADTs). This system enables efficient insertion, retrieval, and analysis of weather data over time, supporting both complete and sparse datasets.



Implementation of Sub Problems:

1. Define the Weather Record ADT with the specified attributes and methods.
2. Implement a 2D array-based storage system for year-wise, city-wise temperature data.
3. Develop row-major and column-major access methods and compare their efficiency.
4. Implement a mechanism to handle sparse data (e.g., sentinel values or sparse matrix).
5. Analyze and document the time and space complexity for key operations (insert, delete, retrieve).

Evaluation Metrics

Metric	Marks	Excellent	Very Good	Satisfactory	Poor
Problem Understanding	1	1	0.75	0.5	0.25
Problem Solving Approach	1	1	0.75	0.5	0.25
Completion of the Problem	1	1	0.75	0.5	0.25
Participation during Class/Lab	1	1	0.75	0.5	0.25
Explanation of the Assignment	0.5	0.5	0.4	0.25	0.1
Attendance	0.5	0.5	0.4	0.25	0.1

Assignment Objectives

- a. Develop a foundational understanding of C++/Python programming and data structures.
- b. Gain practical experience with 2D arrays and abstract data types (ADTs).
- c. Implement real-world applications for organizing and retrieving tabular data.
- d. Learn to handle sparse data and analyze time and space complexity.
- e. Understand memory-efficient data storage techniques.

Assignment Instructions

1. Weather Record ADT Design

Attributes:

- a. **Date:** String or custom structure (e.g., day/month/year).



- b. **City:** String, name of the city.
- c. **Temperature:** Double, temperature value for the city on the given date.

Methods:

- a. **insert(data):** Insert a new weather record into the system.
- b. **delete(data):** Remove a weather record based on specified criteria (e.g., city and date).
- c. **retrieve(city, year):** Retrieve temperature data for a specific city and year.

2. Data Storage Class

Attributes:

- a. 2D array to store temperature values, with rows representing years and columns representing cities.

Methods:

- a. **populateArray():** Populate the 2D array with temperature values.
- b. **rowMajorAccess():** Implement row-major representation for accessing data.
- c. **columnMajorAccess():** Implement column-major representation for accessing data.
- d. **handleSparseData():** Manage scenarios with missing data (e.g., using sentinel values or sparse matrix representation).
- e. **analyzeComplexity():** Provide time and space complexity analysis for storage and retrieval operations.

Evaluation Criteria

Criteria	Marks	Description
Problem Understanding	1	Demonstrates clear understanding of the problem requirements and objectives.
Problem Solving Approach	1	Effective approach to designing and implementing linked lists, stacks, and queues.
Completion of the Problem	1	Complete implementation of all required components (linked lists, stacks, queues, polynomial operations, postfix evaluation).
Participation during	0.5	Active engagement in class/lab discussions and activities



Class/Lab		related to the assignment.
Explanation of the Assignment	1	Clear and detailed explanation of the implemented solution in the submission report.
Attendance	0.5	Consistent attendance in classes/labs relevant to the assignment.

Prerequisites

- Basics of C++/Python programming.
- Understanding of variables, loops, and functions.

Learning Outcomes

1. Efficient organization and retrieval of large tabular data.
2. Understanding of memory-efficient data storage techniques.
3. Ability to implement and compare row-major and column-major data representations.

Submission Guidelines

1. Upload the assignment on GitHub and provide the link of submission by the specified deadline.
2. Ensure code is original and follows the provided format.
3. Include a detailed report covering:
 - a. Description of the Weather Record ADT.
 - b. Strategy for memory representation (row-major vs. column-major).
 - c. Approach to handling sparse data.
 - d. Time and space complexity analysis for key operations.

Helpful references

1. Books:

- a. "Data Structures and Algorithm Analysis in C++" by Mark Allen Weiss
- b. "Python Data Structures and Algorithms" by Benjamin Baka

2. Tutorials:

- a. <https://www.geeksforgeeks.org/sparse-matrix-representations-set-1-using-arrays-and-linked-lists/>



K.R. MANGALAM UNIVERSITY
THE COMPLETE WORLD OF EDUCATION

- b. https://www.w3schools.com/cpp/cpp_arrays.asp