Student Management System (SMS) Project Overview

- Data Structures & Algorithms Implementation
- Professor: Askari
- CS631

- Mohamed Saleh
- 100784454

- G.Sambhavu
- 100818281

1. Data Model: Student

- Fields:
  - – id: unique key (int)
  - – name: std::string
  - – age: int
  - – grade: double
- Serialization:
  - – writeTo(...) writes binary data
  - – readFrom(...) reads binary data

2. Core Container: Binary Search Tree

- Fast insert/search/delete: O(log n) avg
- In-order traversal → sorted by id
- Node struct:
  - struct Node { Student data; Node *L, *R; };

3. CRUD Operations on BST

- Insert (add): recursive, create new node at leaf
- Search (findById): traverse left/right, return pointer
- Delete (removeById):
  - – 0/1 child: replace node
  - – 2 children: swap with inorder successor

4. Display & Sorting

- By ID: in-order traversal into vector, print → sorted by id

- By Grade: sort vector with std::sort → O(n log n)

5. Persistence: Save & Load

- saveToFile: preorder traversal, writeTo(...) each node

- loadFromFile: readFrom(...) until EOF, add() to BST

6. User Interface & Main Loop

- Menu-driven while loop: commands 1–7

- Robust input: clear & ignore on failure

- Map commands to SMS methods: Add, Display, Search, Update, Delete, Save & Exit

7. Error Handling & Flow

- After each std::cin >>: check, clear flags, ignore rest of line

- Flow: Load data → Main menu → CRUD ops → Save on exit

* STUDENT MANAGEMENT SYSTEM :-

- Code Analysis :-

- struct Student
    - ↳ Details about Student
    - ↳ function 'writeTo'
        - ↳ Saves info about student
    - ↳ Calling 'readfrom' function from 'Student'

- class SMS
    - ↳ Node → Access the struct 'Student'
    - ↳ Node 'insert'
        - ↳ Adds User data about Student
    - ↳ Node 'findMin'
        - ↳ Finds the Minimum value as per grade of student.

↳ Node 'remove'
  ↳ Deletes Student

↳ Calling 'search' from 'Student'

↳ function 'inorder'
  ~~↳ Organizes Students~~
  ~~↳ We are using BST~~
  ↳ Defined traversal for BST

↳ function 'savePreOrder'
  ↳ Defined traversal for BST

↳ function 'destroy'
  ↳ Deletes BST to free memory

public:

~SMS() {.....} // destroys BST

void add(.....) // adds Student

bool removeByID(:-) // Removes Student by ID

Student* findByID(...) // Pointer to find a Student

Vector<Student> getAll // A container which stores all Student

function displaySortedbyID()

// Display Students by ID.

function saveToFile(..) // Saves Students list.

function loadfromFile(...)
// loads Students data

main()

{

To show all the text
in Terminal

& the function called.

}