

WIA 1002 Group Assignment 3(16/6/2025)

1. Group Members and Module Assignments:

Group K2-7

- Gan Shu Xian (24004577): Student Records Storage
- Lee Yu Xuan (23098783): Search Engine
- Teoh En Xi (24004582): Sorting
- Yeong Hui Ni (24004529): Recursion
- Tham Wing Shan (24059824): User Interface and Integration
- Kwek Chee Ling (23080328): User Interface and Integration

2. Project Overview:

The **FSKTM Student Enrollment System (SES-FSKTM)** is a Java-based console application designed to manage student data using structured data algorithms efficiently. The system allows the user to insert, delete, search, sort, and analyse student records. Key concepts such as **Binary Search Trees**, **Recursion**, **Sorting Algorithms**, **Search**, and **File I/O** are implemented across five modules, each handled by a specific team member. The data is persisted via CSV files and supports undo insertion operations.

3. Module Descriptions:

- Module 1: Student Records Storage (Gan Shu Xian)
 - Implements a Binary Search Tree(BST) structure to store each student's information, including:
 - Matric ID
 - Name
 - Address
 - Programme
 - CGPA
 - UUID
 - Supports:
 - Insertion
 - Deletion
 - Searching by Matric ID
 - In-order traversal
 - Each node also maintains a subtree size for optimized recursive queries.
 - Ensure average $O(\log n)$ operation time complexity

- Module 2: Search Engine (Lee Yu Xuan)
 - Implements multiple search functions to locate students using:
 - Matric ID
 - Full Name
 - Partial Name (Prefix-based)
 - Categorical Search (e.g. all students from a state or in a programme)
 - Range-based CGPA search
 - Utilises recursion and traversal techniques on the BST to gather filtered results efficiently

- Module 3: Sorting (Teoh En Xi)
 - Implements two different sorting algorithms (eg. Merge Sort and Bubble Sort)
 - Capable of sorting based on:
 - Matric ID
 - Name
 - Programme
 - CGPA
 - Address (State, District, Postcode)
 - Multi-level sorting: allows the user to sort by more than one field (eg by Programme then CGPA)
 - Includes performance comparison based on execution time for both algorithms

- Module 4: Recursion (Yeong Hui Ni)
 - Implements recursive functions to calculate:
 - Total number of students
 - Number of students by programme and by state
 - Number of students within a CGPA range
 - Average CGPA by programme/state
 - Height of the BST
 - Balance factor of any node
 - Demonstrates understanding of tree structure analysis and statistics extraction through recursion

- Module 5: User Interface and Integration (Tham Wing Shan & Kwek Chee Ling)
 - Designs a text-based menu system to allow interaction with all modules
 - Includes:
 - Menu Navigation
 - Data validation
 - File loading/ saving from CSV
 - Duplicate handling during load
 - Stack-based Undo feature for last insertion
 - Handles the full integration of modules for a functional and cohesive application

4. Summary of ChatGPT Usage and Prompts

- Module 1: Student Records Storage
Handled by: Gan Shu Xian (24004577)
Prompts used:
 - Since I need to combine it with other codes, I thought it is highly related to module 5. How will it work?
 - My method has an error and at the same time give a simple code.

How ChatGPT helped:

 - ChatGPT explained the relation between modules to me. In addition, it helps me to check the code and identify which part is wrong. So, a more perfect version is produced.
- Module 2: Search Engine
Handled by: Lee Yu Xuan (23098783)
Prompts used:
 - "Can you explain the search using recursion in a simple way?"
 - "Can you give a simpler version without recursion?"

How ChatGPT helped:

ChatGPT helped me understand tree traversal to search for different student information. It explained that I should start from the root, go left and right using recursion, and check the condition (like name or CGPA) at each node. Later, I asked for a simpler version without recursion.
- Module 3: Sorting

Handle by: Teoh En Xi (24004582)

Prompts used:

- “How to let users choose which fields to sort by and the order?”
- “Can you show a simple way to sort a list of students by multiple fields?”

How ChatGPT helped:

ChatGPT helped me create a menu that allows users to choose which fields to sort (e.g. Matric ID, name, state, programme, CGPA) and whether to sort each one in ascending or descending order.

- Module 4: Recursion

Handle by: Yeong Hui Ni (24004529)

Prompts Used:

- “How do I use recursion to count students by CGPA in a binary search tree?”
- “Can you write a recursive function to count students in Java?”

How ChatGPT helped:

ChatGPT helped me understand how to use recursive tree traversal to calculate statistics like:

- The number of students within a CGPA range
- The number of students from a specific programme
- Support for multiple programme types (AI, SE, MM)
- The height of the tree and balance factor

- Module 5: User Interface and Integration

Handled by: Tham Wing Shan (24059842) & Kwek Chee Ling (23080328)

Prompts used:

- How to build a Java menu-based program that calls multiple modules?
- How to load and save student records from/to CSV?
- How to implement undo feature in Java using Stack?

How ChatGPT helped:

- Provided working structure for switch-case menu navigation
- Explained use of File, Scanner and PrintWriter for CSV I/O
- Suggested use of Stack for undoing last insertion.

5. Software Testing and Debugging

- Testing Strategy:
 - Unit Testing:

- Manually tested insert, delete and search using console inputs
 - Verified each module individually before integrating
- Integration Testing:
 - Loaded actual student records from CSV and tested all menu options
 - Verified file saving and reloading produces the same BST structure
- Sorting Validation:
 - Compared MergeSort and BubbleSort output for consistency
 - Used timing code to validate which algorithm was faster on larger data
- Search Accuracy:
 - Queried for various names, states and CGPA ranges
 - Tested edge cases like empty strings and invalid inputs
- Undo Feature Testing:
 - Inserted students and undid actions to verify proper rollback
- Error Handling:
 - Tested with incorrect file names, malformed CSV entries and invalid inputs
- Debugging Examples:
 - Fixed subtree size not updating correctly by reviewing recursion with ChatGPT
 - Corrected sorting comparator chaining logic after ChatGPT explanation
 - Debugged CSV parser when extra commas in address caused *ArrayIndexOutOfBoundsException*