**Group Assignment 3: FSKTM Student Enrollment System**

| | |
|---|---|
| Course | WIA1002 / WIB1002 Data Structure |
| Topics | Recursion, search and sort, binary search trees |
| Submission format | One zip file containing:<br>- One Java file named FSKTMStudentEnrollment.java<br>- Readme file containing:<br>  ○ Group members & module assignments<br>  ○ Description of each module's functionality<br>  ○ Challenges faced and how ChatGPT helped resolve them<br>- Word document for individual ChatGPT Reflections for each student |
| Group size | 5 students<br>(If your group has fewer or more members, the group leader must distribute the work fairly to ensure equal contribution from each member.) |
| Time limit | 90 minutes, inclusive of time for submitting the file to Spectrum |

**Building a Student Enrollment System for FSKTM**

Your team is tasked with building a student enrolment system for FSKTM: SES-FSKTM. The system must support managing student enrollments using efficient data structures and algorithms.

**Objectives**

Your group will build the system that consists of the following modules:

| Module | Description | Key topic |
|---|---|---|
| Student 1: Student records storage | Implement a Binary Search Tree (BST) where each node stores student info: Matric ID, Name, Address (*break into Street, City, State, District, Postcode for more granular data*), Programme, CGPA and a unique userID (auto-generated, UUID)<br>• Each node should also store subtree size to allow efficient recursive queries for count.<br>• Implement insert, delete, search, and in-order traversal with guaranteed O(log n) average and worst-case time complexity for core operations. | Binary search trees |
| Student 2: Search engine | Implement search functions to search students by:<br>• matric ID<br>• name<br>• partial name matching (prefix search)<br>• categorical matching (i.e. "search for all students from Pahang", "search for all students applying for AI")<br>• range searching (i.e. "search for all students with CGPA between 3.50 and 3.99") | Search |
| Student 3: Sorting | Implement two sorting algorithms to sort patient records by Matric ID, Name, Address, Programme, and CGPA.<br>Note: Address should be sortable by State, District, and Postcode. | Sorting |

| | | |
|---|---|---|
| | • Implement multi-field sorting: i.e. first sort by Matric ID in ascending order, then by Name alphabetically. The user should be able to choose what category/categories and what order (ascending/descending) to sort them.<br>• Add a functionality to compare the performance (time) and speed of both sorting algorithms. | |
| Student 4: Recursion | Use recursion to calculate the following:<br>• Total number of students<br>• Total number of students for each programme<br>Note: Programmes are AI, SE, IS, MM, and CS<br>• Total number of students from each state (i.e. Pahang, Kedah, Selangor).<br>• Total number of students that fulfils a range of CGPA; i.e. user sets a minimum and maximum range, and returns the number of students whose CGPA is inside that range.<br>• Calculate the average CGPA for students in a specific programme or state recursively.<br>• Determine the height of the BST.<br>• Calculate the balance factor at the root or for a specific node (demonstrating understanding of tree balance). | Recursion |
| Student 5: User Interface and Integration | Design a text-based menu system that:<br>• Loads patient data from CSV at startup<br>• Saves data to a CSV file on exit. Able to handle duplicate records whether to overwrite or create a new file. If so, then how to track these files.<br>• Handles integration of all modules<br>• Implements basic input validation and error handling (i.e. checks for duplicate records)<br>• Implements a "history" or "undo" feature for at least one operation (e.g., last insertion or deletion), demonstrating a conceptual understanding of stack-based undo/redo. | UI and Integration |

**AI Coding Assistance**

Students are allowed to use ChatGPT to:

- Explain algorithm behavior.
- Understand underlying concepts.

Students are not allowed to:

- Copy-paste full module solutions without understanding.
- Copy-paste code generated from ChatGPT
- Submit code that they cannot explain or defend.
- Ask ChatGPT to generate the entire project.
- Use Cursor, Augment Code, Replit, Claude and other AI-text editor.
- Use Agent-based / Agentic AI coder

If caught, a mark deduction is applied for the whole group.

**Documentation**

Each group must submit one Readme file containing the following:

- Team members and module assignments. For example:
  - Student 1: [Name] [Matric ID]: student records storage
  - Student 2: [Name] [Matric ID]: search engine
  - ...
- Description of project overview and individual module functions.
- Summary of ChatGPT usage and prompts for each module.
- Software testing and debugging.

Inside the FSKTMStudentEnrollment.java file, each code segment must be clearly labelled which module it belongs to. For example:

// Module 1

...

// Module 2

...

Each student must submit a reflection answering the following questions:

- What prompts did you use?
- How did ChatGPT assist you? Include code snippets and/or screenshots.
- What changes or corrections did you apply?
- Compare the algorithms before and after implementing the corrections. What changed and why?
- What did you learn from the process?

**Suggested Time Allocation**

15 minutes: planning and coordination

16-45 minutes: each student works on assigned module, using ChatGPT for coding assistance.

46-75 minutes: module integration, testing, and debugging

76-90 minutes: complete the documentation for the Readme and Reflections files; submit the assignment to Spectrum

**Grading Rubric**: 10%

Modules 1 – 4: 1.5% each, total 6%

Integration (main method): 1.5%

Readme documentation: 1%

ChatGPT reflections: 1.5%

**Sample Outputs**:

**Module 1: Student Records Storage**

*Inserting new records

```
Adding new student...
Name: Liew Wei Shiung
Matric ID: WHA10061
Street 1: 123, Level 4, Apartment FiveSixSeven
Street 2: Jalan Lapan
City: Petaling Jaya
District: Petaling
State: Selangor
Postcode: 47800
Programme: Artificial Intelligence
CGPA: 2.80
Subtree sizes updated. Current BST height: 3
```

*Deleting student record

```
Enter student matric ID to delete: WHA10061
Deleting...
Matric ID: WHA10061, Name: Liew Wei Shiung, Programme: AI, CGPA: 2.80
Subtree sizes updated. Current BST height: 2
```

**Module 2: Searching**

*Query searching

```
Searching for students applying for AI:
Results: 2 students found
    - Matric ID: WHA10061, Name: Liew Wei Shiung, Programme: AI, CGPA: 2.80
    - Matric ID: WHA10072, Name: Mohd Aladdin, Programme: AI, CGPA: 3.99

Searching for students with CGPA between 3.50 to 3.99:
Results: 1 student found
    - Matric ID: WHA10072, Name: Mohd Aladdin, Programme: AI, CGPA: 3.99

Searching for students by name: “liew”
Results: 2 students found
    - Matric ID: WHA10061, Name: Liew Wei Shiung, Programme: AI, CGPA: 2.80
    - Matric ID: WHA10142, Name: Wong Feng Liew, Programme: SE, CGPA: 3.80
```

*Student not found

```
Enter student name to search for: lew
Search ended. Student not found.
```

**Module 3: Sorting**

*User can choose what field and what order to sort

Choose which fields to sort. Current selection [None]

1. Matric ID
2. Name
3. Address
4. Programme
5. CGPA

You have selected [CGPA]. Which order to sort?

1. Ascending
2. Descending

Choose which fields to sort. Current selection [CGPA] (Descending)

1. Matric ID
2. Name
3. Address
4. Programme
5. CGPA

...

Choose which fields to sort. Current selection [CGPA] (Descending), [Matric ID] (Ascending)

1. Matric ID
2. Name
3. Address
4. Programme
5. CGPA

*Multi-field sorting

Sorted student list by [Matric ID] (Ascending) then [CGPA] (Descending)

- Matric ID: WHA10072, Name: Mohd Aladdin, Programme: AI, CGPA: 3.99
- Matric ID: WHA10081, Name: TestStudentPleaseIgnore, Programme: AI, CGPA: 3.98

**Module 4: Recursion**

```
Students with CGPA between 3.50 and 3.99: 55
Students who apply for AI: 19
Students who apply for AI and SE: 39
Students who apply for AI, SE, and MM: 44

BST height: 5
Balance factor at root: -1 (Right-heavy)
```

**Module 5: UI and Main Method**

*Main menu

```
Welcome to FSKTM Student Enrolment System
------------------------------------------------------------
1. Save/Load Records
2. Insert Student
3. Delete Student
4. Search Student
5. Sort Student Records
6. Show Statistics
7. History
7. Exit

Enter choice: 4
Search by: 1 (ID), 2 (Name), 3 (Partial Name), 4 (Address), 5 (Programme), 6 (CGPA): 3
Enter partial name: liew
Results: 2 students found
-       Matric ID: WHA10061, Name: Liew Wei Shiung, Programme: AI, CGPA: 2.80
-       Matric ID: WHA10142, Name: Wong Feng Liew, Programme: SE, CGPA: 3.80
```

*Loading/Saving Records

```
*Loading from file
Loaded 149 students from file: UG_2025_Sem1.csv
Added 149 students.

*Loading again
Loaded 159 students from file: UG_2025_Sem1_v2.csv
There are 149 duplicate records. Added 10 students.

*Saving data to file
Saving 159 students to: UG_2025_Sem1_v3.csv
```

*Input Validation

Search by: 1 (ID), 2 (Name), 3 (Partial Name), 4 (Address), 5 (Programme), 6 (CGPA): 5
Enter programme to search for: BS
Error: no such programme

Search by: 1 (ID), 2 (Name), 3 (Partial Name), 4 (Address), 5 (Programme), 6 (CGPA): 6
Enter CGPA range to search for (min),(max): 3.99
Error: please enter a valid range to search for: 3.99,-1.99
Error: please enter a valid range to search for: 3.99,1.99
Searching for students with CGPA between (1.99) and (3.99):
Results: 21 students found

- Matric ID: WHA10061, Name: Liew Wei Shiung, Programme: AI, CGPA: 2.80
- ...