

TUTOR MANAGEMENT SYSTEM

eXcel Tuition Centre is one of the fast-growing tuition Centre in Malaysia that employed tens of tutor to conduct tuition classes on all the school subjects in their various Tuition Centers. All the tutors are paid by hourly rate ranged from RM40.00 – RM80.00, depends on the subject they teach. Tutors can be rated by the students on their teaching performance with a scale rating from 1-5, where 1 is “very poor performance” and 5 is “excellent performance”. The tuition Centre needs a system to store information about their tutors (identified by Tutor ID, Name, Date Joined, Date Terminated, Hourly Rate, Phone, Address, Tuition Centre Code, Tuition Center Name, Subject Code, Subject Name, Rating [a single digit integer number ranging from 1 to 5]). To maintain quality and consistency, one tutor is allowed to teach on the subject in their specialized field only.

Tuition Centres are located in various locations and each of the Tuition Centre is managed by an admin staff. All the tutors’ record are managed and maintained by the Tuition Centre Head Quarter Human Resources Department. The current filing system of records is by categorizing the tutors by their respective location. i.e. all tutors’ records are filed together in the tuition centre location. In each location, the tutor’s records are filed alphabetically by the tutor’s name. The main problem with this filing system is when searching for a record, the record could not be easily found as it might have been misplaced in another Tuition Centre or the record is not placed in the correct alphabet sequence. This problem proves challenging for the Tuition Center admin staff when wanting to retrieve a tutor’s record quickly. When a tutor terminated his tutoring service, the record for that tutor will be destroyed 6 months after the termination.

At the end of each week, a report is generated to list all tutors of the Tuition Center to keep track of the available vacancy for new tutors.

The admin manager of the tuition center has decided that it is time to computerize the records of Tutors serving in the Tuition Center due to the problems with the manual filling system. Since you have some knowledge in developing a computerized system, the Tuition Center HR Manager has approaches you to assist them in developing the system.

This project requires you to develop C++ program for the Tutors Management System which should contains features stated below:

- i. Add a Tutor Record
 - ii. Display All Records
 - iii. Search a Tutor by Tutor ID
 - iv. Search Tutors by overall performance (Rating)
 - v. Sort and display by Tutors ID in ascending order
 - vi. Sort and display by Tutors Hourly Pay Rate in ascending order
 - vii. Sort and display by Tutors Overall Performance in ascending order
 - viii. Modify a Tutor Record
 - ix. Delete a Tutor Record
 - x. Exit
- a) You are required to develop the system using **TWO** approaches:
- i. **Array of Structure**
 - ii. **Linked List**

- b) The list of TUTORS should contain the following information:
Tutor ID, Name, Data Joined, Date Terminated, Hourly Pay Rate, Phone, Address, Tuition Center Code, Tuition Center Name, Subject Code, Subject Name and Rating.
- c) **You must pre-define some tutors listing in the system.** Besides, your “**Add a Tutor Record**” function in your menu will allow the user to enter new Tutor.
- d) Your program should also allow the user to search for a particular type of Tutor. For eg: a search for Tutor in “Bukit Jalil” should only display listings of tutors serving Bukit Jalil’s Tuition Center. **Note that you must apply some suitable searching algorithms in this section.**
- e) When displaying the tutor, the user must be able to move back and forth between the lists of tutor. Listing of tutors must be sorted as mentioned above. **Note that you must apply some suitable sorting algorithms.**
- f) The HR Manager should be able to delete or modify the record of a tutor. **However, the record of a Tutor can be deleted 6 months after leaving the Tuition Center.** *Note that only the Tutor phone and address attributes may be modified.*
- g) Include all validations required for the system and use good programming practices (eg. indentation, meaningful identifier names, comments, and etc.).
- h) You must compare the TWO approaches (Array of Structure and Linked List) in their strength and weaknesses in implementation the system.

Assignment Requirements

This is a group assignment, which comprises of **THREE (3)** members.

You are required to submit a **softcopy of assignment report** (*refer to appendix 2 for more details*) **and source code**.

Note: Approximation of total pages for the final report: 20 - 50 (max).

The final report should contain:

- Detailed explanation of the structure data structures created, with proper justification on your decisions (include source code defining structure, data structure, data members and etc).
- Brief explanation about the algorithms used to implement the functionalities stated above (include code snippets of important parts of implementation).
- Source code of the main function, with screenshots showing program's input and output interactions.
- Workload matrix or contribution of each team member in the assignment must be include in the documentation, and signoff by each of the team member. Distribution of work across components such as design, implementation, documentation and presentation ideally equal between members, otherwise, marks will be reflected based on the contribution.

Besides, you need to submit your proposal on the **LMS in week 8**. The proposal should include the proposed data structure together with related algorithms (*refer to appendix 1 for more details*). The algorithms should be presented in **flowchart or pseudocode**. This proposal will contribute **20%** for the final assignment marks (under Design criteria).

You have to **present your assignment solution and answers** to the lecturer during a Q&A session that will be conducted after the hand-in date.

If you use some code which has been taken or adapted from another source (book, magazine, internet, forum, etc.) then this must be **cited and referenced** using **APA Referencing Style within your source code**, and this must be mentioned explicitly in the **report**. Failure to reference code properly will be treated as plagiarism. **Automated tools for checking code similarities** among submissions will be used, and all detected cases will be treated as cheating. Assessment marks are divided as follows:

Total Percentage	80%		20%	
	Design (proposal)	Implementation	Documentation	Presentation
Total raw marks in each section	25	75	50	50

What You Need to Hand In?

1. You are required to hand in the individual assignment report on or before the due date mentioned on the cover sheet of the assignment in the APU Learning Management System (APSpace).
2. A softcopy of the report (*in Word Document format*), in addition to the **C++ files** of the programs. The organization of files and folders **must adhere to the following instructions precisely**:

- The report should be named using format
“<GroupNo>_<student ID-leader>_<student ID-member1>_<student ID-member2>.docx”.

For example “**G1_TP012345_TP012344_TP012123.docx**”

- All the source codes (**.cpp and .h**) should be zipped into one file and named following the above format. Make sure to **DELETE** all non-source-code files, including executables (*.exe).
3. You should **present an executable solution** during Q&A session to demonstrate program execution, the working of the data structure, your understanding of the code, and ability to modify / fix it.

Marking Criteria:

The program submitted will be evaluated according to the following performance criteria:

Distinction (90% and above)

- Program compiles and executes perfectly
- At least 90% of the required functionalities are correctly implemented
- Efficient data structures and/or algorithms are used in the implementation
- Clear coding style and structure, and code is properly commented
- Functionalities are fully tested/validated in program execution

Credit (70% – 89%)

- Program compiles and executes
- Between 70% and 90% of the required functionalities are correctly implemented
- Implementation uses a data structure or algorithm that is not most efficient
- Clear coding style, and code is properly commented
- Functionalities are not fully tested/validated in program execution

Pass (50% - 69%)

- Program compiles perfectly and executes
- Between 50% and 70% of the required functionalities are correctly implemented
- Implementation uses inefficient data structures or algorithms
- Unclear coding style, or code is not properly commented
- Functionalities are not fully tested/validated in program execution, or produce errors in some cases

Marginal Fail (30% - 49%)

- Program does not compile or run, but coding logic is almost correct
- Between 30% and 50% of the required functionalities are correctly implemented
- Implementation uses inefficient data structures or algorithms
- Unclear coding style, and no comments provided
- Functionalities are not tested/validated in program execution

Fail (below 30%)

- Program is not given
- Program does not compile or run
- Less than 30% of the required functionalities are implemented
- Implementation uses very inefficient data structures or algorithms
- No proper code structure and no comments provided

APPENDIX 1: SHORT PROPOSAL OF YOUR GROUP PROJECT:

Note: Approximation of Total Pages for the proposal: 10 - 30 (max).

Before your group start to develop the tuition management system, your group is required to write a short proposal to identify what are the data members will be included your structure, what are workflows involved in the system, and which of the algorithms will be implemented in your future system.

Below are the minimum elements that you need to include in your proposal:

1. Introduce your group's members.
2. Identify what are the important data structures or classes (optional) will be included in your system.
 - Detailed explanation of the data structures and classes created, with proper justification on your decisions (include source code defining classes, data members, and method headers only).
3. Briefly explain all of the workflows planning for your future system.
 - Use flowchart / pseudocode to describe each of the functionalities' workflows.
 - Provide a brief explanation about the algorithms that you plan to use in your function
 - Provide a proper justification on your decisions.
4. Provide a workload distribution table to determine the tasks of the members.

You are required to submit a softcopy of proposal to the LMS before week 8. If your group fail to submit the proposal report, it will cause your group to get the 0 mark from your design section (20%).

Assessment marks in design section are divided as follows:

Total Percentage	Design Section (20%)	
	Data Structure (design)	System Workflow and Algorithm
Total raw marks in each section	10 Marks	15 Marks

Sample Outline of the Proposal:

The proposal outline as below:

- Cover Page
 - Module Code and Name
 - Intake code
 - Proposal Title
 - Include your group member list.
- Introduction
 - Brief introduction about the system
 - Detailed explanation of the data structures and classes created
 - Provide justification on your decisions
- System Design and Workflows
 - System workflow diagram (Flowchart / Pseudocode)
 - Brief explanation about the algorithms
 - Provide justification on your decisions
- Summary table of the task distribution between the group members.

APPENDIX 2: OUTLINE OF YOUR FINAL REPORT:

The final report outline as below:

- Cover Page
 - Module Code and Name
 - Intake code
 - Proposal Title
 - Include your group member list.
- Introduction
 - Brief introduction about the system
 - Source code of each developed data structure / class
 - Detailed explanation of the data structures and classes created
 - Provide justification on your decisions
- Implementation:
 - Source code of each algorithm.
 - Brief explanation about the algorithms
 - Provide justification on your decisions
- Result
 - System Input / Output Screenshot
 - Brief explanation about screenshots
 - Strength and weaknesses of the proposed approaches
- Conclusion, Future Works and Reflection.
 - A brief summary about your system (including limitations)
 - Brief description of your future works based on your system limitations
 - Briefly describe your experience/feedback on the assignment
- References
- Appendix: Workload Matrix Table with signature