AACS1084 PROGRAMMING CONCEPTS AND DESIGN II
ASSIGNMENT

Students are to work in a team of 4 to 5 members.

Weighting towards the coursework: 50%

Assignment Overview	Your team's task is to design and build a console-based system using C language. The requirement is to develop a system that can be used to maintain recreation facilities for a club. The system should contain a selection of modules from the following list: • Staff Information Module – to add staff login account and maintain staff login details. • Facility Module – to record the details of facilities, such as karaoke room, snooker table, gymnasium, table tennis, squash courts etc. • Member Information Module – to maintain information about member details. In order to use the facility, one must register as a member. • Booking Module – to book the facility at least one day in advance. • Facility Usage Module – to record facilities' usage information. From the module, it also allows the staff to view the facilities usage status. Based on the given list, each member is required to choose and be in charge of **ONE** module. The number of modules in the system should be based on the number of members in your team. You are required to research on the functionalities and logic flow of the module which you are in charge.
Learning Outcomes Being Assessed	CLO3: Apply principles and techniques for good program design, modularization, structures and files in developing programs (C3,PLO2)
Submission Deadlines	**Assignment progress** – by **Week 7**, your team must have confirmed each member's tasks in the assignment, in particular the module in charge and type of file used. Your tutor may also check on your structure chart design etc. **Presentation and Final submission of Report** – Presentation is on **week 11/12**. By Week 10, all the modules by team members should be integrated, normally by using a main menu. In Week 11/12 during the classes, each team member will present (demonstrate/demo) his/her part of the system. The assignment final submission of report is to be submitted on **week 11**. Your submission should include: 1. The integrated console project. 2. Softcopy of the complete assignment report. **[Note: For late submission, there will be a reduction of absolute marks from the mark's score submitted:]** • Late 1 to 3 days after deadline of submission: minus 10 marks; • Late 4 to 7 days after deadline of submission: minus 20 marks; • Late more than 7 days after deadline of submission: 0 marks

Assignment Details

1. **MODULES**.

   As stated in “Assignment Overview”, each team member is required to choose and be in charge of **ONE** module.

   Your module must involve a file with **at least 6 data fields**. You are encouraged to add in more data fields in order to enhance the application’s logic and practicality.

   Examples of data fields are listed below. Add a few on your own. For counting purposes, date and time will each be taken as one field (even though they consist of 2 or more subfields)

- **Staff Information Module**
  - Staff ID, name, password, password recovery, position, etc.
  - E.g.: ST0001, Patrick Teoh, 1234, numbers, Administrator, …

- **Facility Module**
  - Facility ID, types, description, venue, maximum allowable users, etc.
  - E.g.:KA1, Karaoke, Karaoke room1, Main Block, 10, …
    (Note Facility ID comprises a 2-letter code and a digit eg. KA1,KA2 for karaoke room 1 and 2. ST1,ST2 for snooker table 1 & 2 etc.)

- **Member Information Module**
  - Member ID , name, gender, IC, contact number,etc.
  - E.g.: M1001, Chin Ting Wei, F, 881205-14-1234, 013-123 4567, …

- **Booking Module**
  - Booking ID, today’s date, booking date, booking time, member ID, facility ID,etc.
  - E.g. B1116, 5/11/2019, 10/11/2019, 10am-11am, S1001, KA1, …

- **Facility Usage Module**
  - Today’s date, time, member ID, facility ID, usage type, etc.
  - E.g.1. 10/11/2019, 10am-11am, 19WMD09123, F0001, booked, …
  - E.g.2. 10/11/2019, 10am-11am, 18WMR20568, F0012, walked-in, …

2. **CONCEPTS INCORPORATED**.

   Each module **must incorporate** the following 3 programming concepts and topics that have been covered in this course:

   - **Structures**
     - Include as many *useful fields* as you feel is necessary
     - Incorporate *nested structure* to show your understanding.

   - **Text file or binary file**
     - In general, each team member is to be in charge of only one file. You need to decide on the file format to use (either text or binary), but your whole team must use an equal (or almost equal) number of text + binary files, ie 2 text files + 2 binary files for a 4-member team, and 2+3 or 3+2 text/binary files for a 5-member team.
     - You are expected to be able to process the files correctly (ie. retrieve / update records).

   - **User-Defined Functions**
     - Enhance efficiency, readability and re-usability by using functions whenever appropriate.
     - Include parameters where appropriate and minimize/eliminate the use of global variables.

3. REQUIREMENTS.

Each module must include these 4 compulsory file functions listed in a menu:

- The **Add** function should save new record(s) into the text/binary file. Some basic data validation should be done before adding (saving) the new record(s).
- The **Search** function should retrieve data based on given criteria from the text/binary file and display it in a suitable format.
- The **Modify** function should allow a user to make changes to the data. Similarly to adding, some basic data validation should be done before saving modified record(s).
- The **Display** function should display all the records in an appropriate tabular format.

[Suggestion: At the start of your module run, read all data from the text/binary file into an array of structures. Perform the necessary processing on the array. At the end of the module run, write the updated array of structures back to the text/binary file.]

In addition to the 4 compulsory functions above, **include at least 2 other functions** of your choice that is useful to the user. As examples, you may have a Reports function and a Delete function.

In terms of teamwork, more consideration will be given if your module is able to use data from other modules. In this case, the team member in charge of the other module should write the appropriate function for your module to call.

4. SYSTEM DEVELOPMENT.

Each team member will write a complete C program for his/her own module (i.e. the main program + necessary functions). The program must be tested as thoroughly as possible before integrating (combining) with other modules. (This is to minimize problems of correcting and re-combining modules.) For a 4-member team, there will now be 4 C Project folders (with 4 main programs). When ready for integration, ***change the respective*** **`main()`** program to a function named after the module, eg. **`void staffMain()`**, **`void staffDisplay()`** etc. Submit a copy of the Project folder to the team leader.

SYSTEM INTEGRATION.

The team leader will create a new Project and write the 'final' main program containing a menu listing the 4 module choices. Then add all the other module files to the Project (use "Add / Existing Item"). Resolve all integration errors, e.g. rename functions, change exit( ) to return etc. The whole team should be involved. You are advised to do this one week before presentation time.

5. PRESENTATION.

Each member must demonstrate the module he/she is in charge of to the respective tutor in week 11/12. Your demonstration shall involve the following:

- **Demo.**
  - **Before demo** – Your file must already contain **at least 10 records** with realistic data. For text files, you may key them in manually. For binary files, do add records to your file using your Add function. All this must be done at least one day before your demo. Ensure you also do a test run on your own.
  - **During demo** - Briefly demonstrate your module. Explain the structure(s), file(s), and functions used if requested to do so.

- **Q&A.** The tutor will check on your understanding of your part of the system and to ensure originality of your work, you may be required to make simple / reasonable on-the-spot code changes.

Assignment Report's Content	(Use the template provided) • Cover Page • DECLARATION OF ORIGINALITY • Assignment Evaluation Form • Table of Contents – List of contents and their page numbers. **1. Introduction -** Description of the assignment system. **2. Overall System / Module Structure Chart** showing the structure of the whole program with at least 3 levels (should fit into 1 page only). **3. Home / Splash screen(s)** (eg. logo/main menu etc) - provide screenshot(s) (black text, white background) and a brief description. **4. System modules** – For this section, each student will contribute a sub-chapter containing their own module documentation. Each subchapter should be in the following format: o 4.1 **< Module Name > by < Student1 name >** o 4.1.**1 Brief Description** Briefly describe the functions available by point form or separated by paragraph. o 4.1.2 **Outputs & File Contents** Briefly describe your planned transactions, followed by narrated screenshots black text, white background) to show and prove the module's functionalities. You should include the screenshots from the beginning to the end of each process. Eg: • Module's main page • File's existing data (for text file) or display all records • Perform a transaction, such as adding or editing a record • Display file's updated data • Perform another transaction and show resulting records, etc. o 4.2 **< Module Name > by < Student2 name >** etc.

<table>
<tr>
<td><b>Report Format</b></td>
<td>
<p>Your assignment report should adhere to the following guidelines.</p>
<table>
<tr><td>Paper size</td><td>A4 (Use only one side of the paper)</td></tr>
<tr><td>Line spacing</td><td>1.5 lines</td></tr>
<tr><td>Font</td><td>11 points. For general words use <b>Times New Roman</b>; for code use <b><code>Courier New</code></b></td></tr>
<tr><td>Alignment</td><td>Text – justify</td></tr>
</table>
</td>
</tr>
<tr>
<td><b>Plagiarism Consequence</b></td>
<td>
<p><b>Any student caught cheating, or whose code is suspected not to be genuinely created by himself/herself, or who submits plagiarized work, must redo and resubmit his/her own work within 7 days.</b></p>
<p><b>If the reproduced work fulfils the assignment requirements, the maximum mark that the student can be awarded is a passing mark, otherwise the student will fail the assignment.</b></p>
<p><b>Similarly, students who allow their friends to copy their assignment code or work will have their own marks downgraded to the passing mark only.</b></p>
</td>
</tr>
<tr>
<td><b>Assessment Criteria</b></td>
<td>
<p>Although this is a team assignment, marks will be given individually based on each member's work.</p>
<table>
<tr><td>Assessment Criteria</td><td>Marks Allocated</td></tr>
<tr><td>• Application of programming knowledge</td><td></td></tr>
<tr><td>○ Structures</td><td>12</td></tr>
<tr><td>○ File processing (Text / binary files)</td><td>12</td></tr>
<tr><td>○ 4 basic file functions</td><td>12</td></tr>
<tr><td>○ Others – file functions, re-usable / shared functions etc</td><td>12</td></tr>
<tr><td>• Teamwork – system integration, report documentation, cooperation etc</td><td>12</td></tr>
<tr><td>• Program originality, efficiency and readability</td><td>10</td></tr>
<tr><td>• Report</td><td>20</td></tr>
<tr><td>• Presentation</td><td>10</td></tr>
<tr><td><b>Total</b></td><td><b>100</b></td></tr>
</table>
<p>In addition to the late penalty, marks will also be deducted for bad programming practices, eg. Using the goto statement or function calling itself (or calling back the caller function etc) instead of using a proper loop, etc</p>
</td>
</tr>
</table>