This is an exercise in file I/O management only. **Do not use a list or array outside of a method or as a property to manage the data, it will not be graded.**

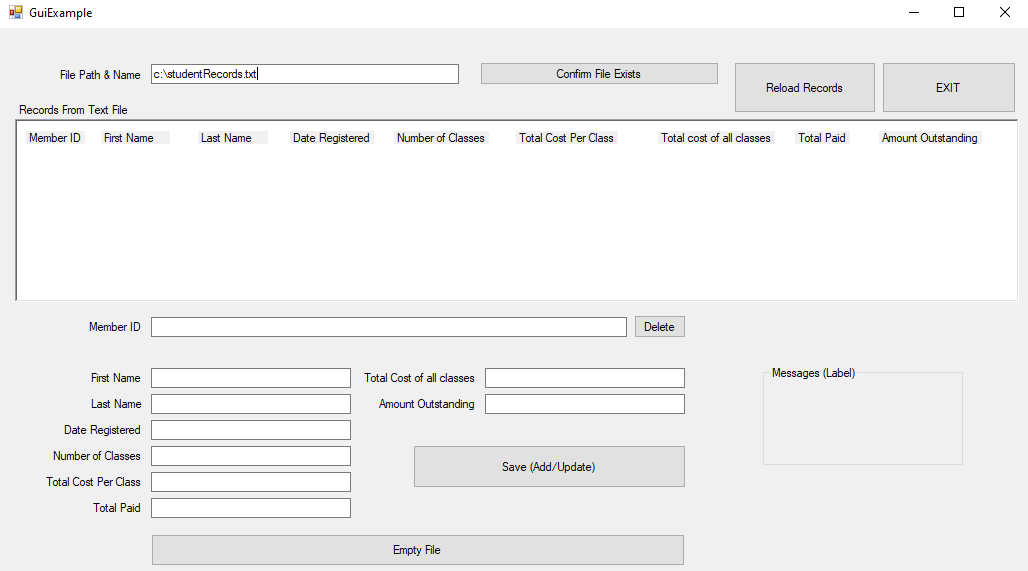
**You program must allow adding, editing, and deleting, multiple records to/from a file (as described in the questions below) – ensure you read the entire question and understand it before writing any code.**

## Assignment 3 – File I/O

Create an application which allows a user to store and edit gym member registration information. The data must be saved in a text file where each row represents a gym member record.

**GUI DESIGN:**

[3 marks] Your screen layout must match the following design:



**REQUIRED FIELDS:**

[2 marks]You are required to create meaningful field names, but all field names must end with the initials of your first name and last name; for example for student John Doe the member ID field could be named “**memberIdJd**” where Jd is required at the end.The following fields must be stored in each record, in the order listed **(also note the format for display where required in program below)**:

1. Member ID –*[1 mark ] Must be displayed with all letters capitalized if included (eg. QD3435FA)*
2. First Name - *[1 mark] First letter of name must be displayed capitalized (eg. Johny)*
3. Last Name – *[1 mark] First letter of name must be displayed capitalized (eg. Brown)*
4. Date registered – *[1 mark] must be displayed in date format*
5. number of classes - *[1 mark] must displayed as a whole number (eg 12)*
6. Total cost per class – *[1 mark] Must be displayed as currency (eg $12,500.56)*
7. Total cost of all classes – **calculated** (number of classes \* total cost per class); [1 mark] *Must be displayed as currency (eg $12,500.56)*
8. Total Paid – *[1 mark] Must be displayed as currency (eg $12,500.56)*
9. Amount Outstanding – **calculated** (Total Paid-Total cost of all classes);*[1 mark] Must be displayed as currency (eg $12,500.56)*

[1 mark] The fields stored in the file must be separated by a semicolon “;” (delimiter).

**RECORD LIST DISPLAY REQUIREMENTS:**

Where required (See add/update/delete questions below) the list of records must be displayed to the screen using **ONE** of the following options:

* [5 marks] Rich Text Box
  + If used all fields must be clearly shown and each record (with all fields as shown above) must be on 1 line.
  + Rich Text box must be read only.
* [5 marks] Grid views
  + Users must not be allowed to Resize Columns or Add Rows to the grid.
  + The grid must be Read Only
  + The Row Headers Visibility should be set to false.

**APPLICATION START:**

The user must create the file before anything can be done (ie. Record add/update/delete)

When the program starts the user must first select “Confirm File exists”. All other options (save, delete, reload records, empty records) must be disabled,

[3 Marks] The “file path and name” field is required - if the user selects “Confirm File exists” and the field is blank display an error stating – the “file path and name” field is required.

[2 marks] if the directory path is not included with the file name (eg C:\) the file must be saved in the project directory.

“Confirm file exists” must check for the file as entered in the “file path and name” field:

1. [2 marks] **If the file does not exist,** create the file, and display a message to the screen notifying the user that a new file has been created.
2. [2 marks] **If the file exists,** the program must list all the existing records to the screen

Note:  
Any new or existing data (during an add, edit, or delete) must be updated on the list shown on screen.

**ADD / UPDATE RECORD:**[2 marks] Create the logic to update or add a new record to the text file.

[2 marks] If the record does not exist in the file (based on the member ID field), insert the record.

[2marks] Otherwise if the record exists update the record with the data the user has entered.

The following **constraints** must be met for each screen field before saving (Add or update) **see display requirements above**:

* 1. [1 mark] **The member ID** - *must be unique it can contain letters and numbers.*
  2. [2 marks] **First name** –   
     *must be more than 2 characters and must not contain numbers*
  3. [2 marks **Last Name** –   
     *must be more than 2 characters and must not contain numbers*
  4. [2 marks] **Date registered** –   
     *must not be in the future*
  5. [1 mark] **number of classes** –   
     *must be greater than or equal to 1; (field must only accept numbers)*
  6. **Total cost per class** –   
     *must be greater than or equal to 1; (field must only accept numbers)*
  7. [1 mark] **Total Paid** –   
     *Must be displayed as currency (eg $12,500.56)*

Data entry fields must not be created for these calculated field below, but the fields must be displayed in the list:

* 1. [2 marks] **Total cost of all classes** (calculated see required fields section above) –   
     displayed and stored in file; *Must be displayed as currency (eg $12,500.56)*
  2. [2 marks] **Amount Outstanding** (calculated see required fields section above) –   
     displayed and stored in file; *Must be displayed as currency (eg $12,500.56)*

**when the user clicks save:**

[3 marks] Any errors (data entry or otherwise) during the insert or update, must be displayed all at once, one line per error, and give the focus to the first field in error.

[3 marks] The record must **not** be saved if the data entered has any errors (see **constraints** above)

[2 marks] If there are no errors the new information must be saved to the text file

[3 marks] If there are no errors the new information must also be displayed/updated in the list.

[2 marks] The screen list must be updated correctly to display all records with no duplication of old records previously displayed.

[1 mark] The fields for all records listed must be displayed in the order as listed above under “required fields” from 1-9.

**DELETE RECORD:**

Create logic to allow user to delete a record

[3 Marks] The user must enter the member ID and click “delete” In order to delete a transaction.

[3 marks] If the member ID for the record to be deleted is not found – display a message to the user “Record to be deleted is not found”.

**RELOAD RECORDS:**

[3 marks] Create logic – If the user selects “Reload Records” reload all records to the grid view or rich text box (see record list display requirements).

[2 marks] Records must not be duplicated.

**EMPTY FILE:**

[3 marks] Selecting “Empty file” must remove all records from the text file and clear the rich text box or grid view.

**EXIT PROGRAM:**

[1 mark] Selecting “Exit” exits the program.

**FILE I/O:**

[3 marks] Catch all file I/O exceptions (open/read/write) and display them as error messages on the form.

**ERROR MESSAGES:**

[3 marks] All error messages must be displayed in the “Messages Area (Use text box or label to display). After the user clicks to complete an action (eg. Delete, Save, Reload Records etc) Clear all old messages before running the action for the user