|  |  |  |
| --- | --- | --- |
| COMP7013 OOP | | |
|  |  |  |
| Practical Assessment 30% | | |

# Assessment structure

|  |  |  |
| --- | --- | --- |
| 1. This assessment is split into 3 stages with 3 different due dates | | |
|  | | * 1. Stage 1 – Friday of week 4 |
|  | | * 1. Stage 2 – Friday of week 6 |
|  | | * 1. Stage 3 – Friday of week 8 |
| 1. Each stage is equally weighted; they are all worth 10% of your final grade | | |
| 1. Submission is through the Canvas Assignments tab | | |
| 1. Submit a zip file with all of your Java code and a screen recording of your program showing all of the requested functionality | | |
|  | * 1. If you do not include a screen recording of your code working then you will have marks deducted from your grade | |
| 1. Any assignments that are submitted late will be subject to the usual MTU late penalties | | |
|  | | * 1. 10% deduction for submissions less than 1 week late |
|  | | * 1. 20% deduction for submissions less than 2 weeks late |

# Project outline – student records GUI

I want you to develop an application that stores student records at MTU; the records consist of information about each student and information about the modules they have completed and the grades they received. This will allow the university to stay informed about student performance.

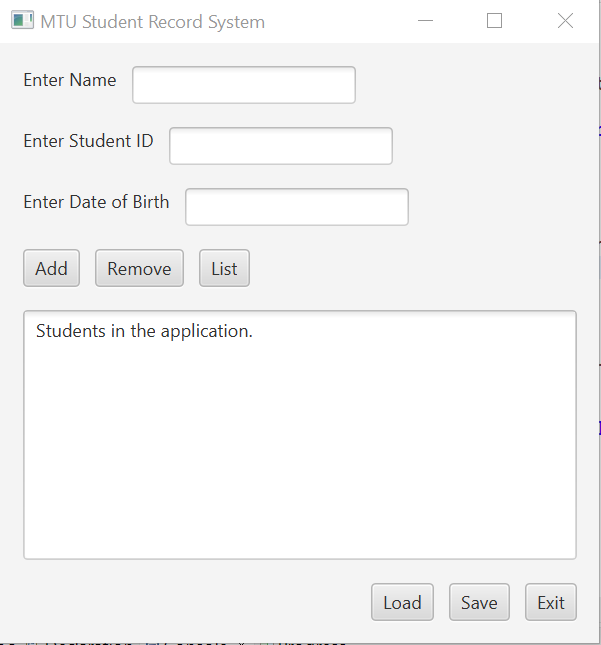
You will use an ArrayList to store Student objects.

Write a JavaFX desktop application with a GUI.

Do not use Scene Builder to build the GUI, this is an exercise in using a pre-existing Java-API.

# Stage 1 – 10 marks

|  |  |
| --- | --- |
|  | 1. Use an ArrayList of objects (Students) to store information inputted using the “Add” button |
|  | 1. Display all information in the text area. This will show the text from the ArrayList of Students. |
|  | 1. “List” updates the list of Students displayed |
|  | 1. A Student can be removed from the ArrayList using the “Remove” button |
|  | 1. “Load” and “Save” do as they imply: load and save the information to or from a text file. Load automatically on startup. |
|  | 1. Ask the user to save or not on exit. |
|  | 1. Use the below GUI as a guide, with some creativity you can improve on it. |



# Stage 2 – 10 marks

|  |  |  |
| --- | --- | --- |
|  | 1. You will need to extend the code you developed for stage 1 to complete this stage | |
|  | 1. Use a TabPane to split your GUI into 3 tabs | |
|  | 1. Tab 1 | |
|  |  | * 1. A tab for adding students, this tab will look like the GUI you built in stage 1 |
|  | 1. Tab 2 | |
|  |  | * 1. Tab for recording the modules each student has completed   2. You will select a student from the ArrayList (this should be selected from a drop-down list or other to reduce the possibility of entry error)   3. Add the name of a module they have completed   4. Add the grade they got in that module |
|  | 1. Tab 3 | |
|  |  | * 1. This tab will be used to view the records for each student   2. The user can select a student and all of their information and the list of modules they have completed will be displayed |
|  | 1. Make sure you use a good layout for the items on each tab, do not make it too simple. | |
|  | 1. MVC must be used, separation between Model, View, and Controller is now required. | |

# Stage 3 – 10 marks

|  |  |
| --- | --- |
|  | 1. You should again build on the previous work that you have done and modify existing code to include these new features. |
|  | 1. The results displayed in tab 3 should be ordered based on the grade attained by the student or by the module name (in alphabetical order). The user can select how the results are ordered using a button, or some other control device. |
|  | 1. Use Serialization to Save the student details using serialization to a file. The same for Loading contact details from a file. |
|  | 1. A good package structure; you should consider how your different classes can be separated into packages to improve code readability. Consider how this will help your MVC design pattern structure. |
|  | 1. Generate a Test Suite with at least two test cases. Examples (but not restricted) of items to test are integrity checks (on name or student id), the sorting of lists of students |