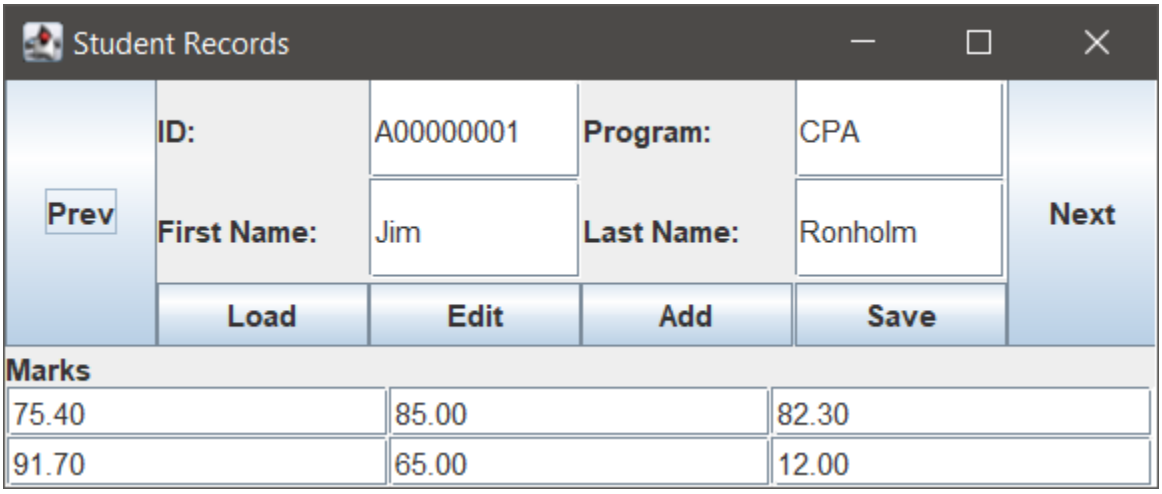
F2020

Assignment 3
Events & Collections

Student Records

Assuming a GUI design similar to the following:



In the first `StudentRecords` assignment you built the gui. In the second you added a backing storage using some sort of `Collection` and you wrote the code to manage the `Prev`, `Next`, `Edit`, and `Add` buttons – however you left out the "`Marks`" functionality.

In this assignment you will complete the application. Your application will now include the `Marks` in the Student record. Additionally, you will complete the `Load` and `Save` functionality as described.

1. When you add or edit a `Student` object to the backing collection you must include the `Marks`.
    - Don't be concerned with validation – the only thing the instructor solution does in this regard is catch `NumberFormatException` if/when the contents can't be converted to double – in which case it just sets that mark to 0.0
2. When the user presses the "`Save`" button the program will save the current records.
    - open a `JFileChooser` to allow the user to specify which a filename to save the current records to
    - save the records
3. When the user presses the "`Load`" button the program will load saved records into the application
    - open a `JFileChooser` to allow the user to specify which file should be opened
    - open the file
    - import the records into your application (replace any existing records)

At all points, every `JButton` should only be enabled when it makes sense. For example, you should not be able to save records if there aren't any! Note: you cannot `Save` or `Load` while a record is being edited.

Assignment 3
Events & Collections

What should your program save to the file? There are a lot of different ways this could be written – you could write individual `StudentRecord` objects, you could write your entire collection with one write, you could use text files, or binary files. No working solution will be considered incorrect/wrong but you should be prepared to defend your reasons for the choices you make. NOTE: you will need to implement `Serializable` on the `Student` class before you will be able to save `Student` objects.

Some additional comments on how this could be a better application

There are a lot of ways this application could be extended – that don't lend themselves well to a school assignment. For example, there is a lot of validation that could be carried out. It's not that we forget about them when working on a project like this – it's just that we have to accept a smaller scope to make the assignment achievable in the amount of time we have.