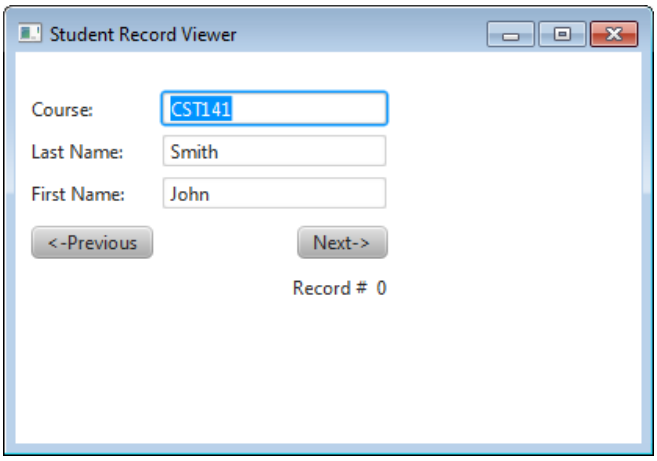
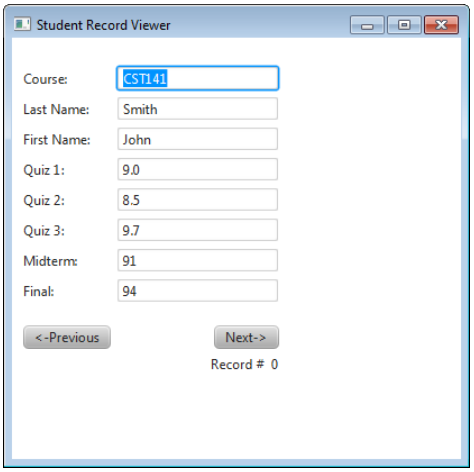
CST141-SP17
Project 03
Due: 5/01 by 11:59PM

Our third project is going to require you to modify an application that reads in records from a file and allows a user to skip forward and backward to view the data in a form. The form is created using JavaFX. Create a new NetBeans JavaFX Application Project called "Project03". Copy the contents of the Project03Source.txt file over the code in the NetBeans editor. Fill in your name in the comment block. Copy the "studentdata.txt" file into the package folder of the src folder for the project. Set the working directory to the package folder. Compile and run the program to confirm you have the application working and you can see how it functions. You should see a screen like the following:



Step 1: Define new Instance Variables for text fields for the remaining fields in each record of the text file. Then add labels and the text fields to the centerPane GridPane. The additional fields are "Quiz 1", "Quiz 2", "Quiz 3", "Midterm", and "Final". Expand the size of height of the stage which is currently set to 250 pixels to 375. Move the buttons and the Record # label to lower rows to accommodate the new fields (rows 11 & 12). Remember the position in a GridPane is specified by column # followed by a comma and then row #. So for instance, "centerPane.add(butPrevious,0,4);" puts the button in column 0 and row 4. Don't forget to add the appropriate new assignment statements to the Project03 No-Argument Constructor and modify the setFields() method to assign values to the fields. Confirm that the changes work before moving on (see example on next page).



Step 2: Add text field declarations to the Instance Variable section, assignments in the Constructor, and add the labels and the fields for "Final Grade" and "Letter Grade" to the form. The setFields() method has to calculate the Final Grade and then the Letter Grade from the Final Grade. This means you must convert String values to double such as "Double.parseDouble(fields[3])". The Quizzes are in indexes 3, 4, and 5 and the Midterm grade and Final grade are in 6 & 7. You can call additional methods if you want to keep the code neat.

Final Grade = (average of 3 quiz grades converted to 100 pt scale * 30%) + (midterm grade * 35%) + (final * 35%) rounded with Math.round().

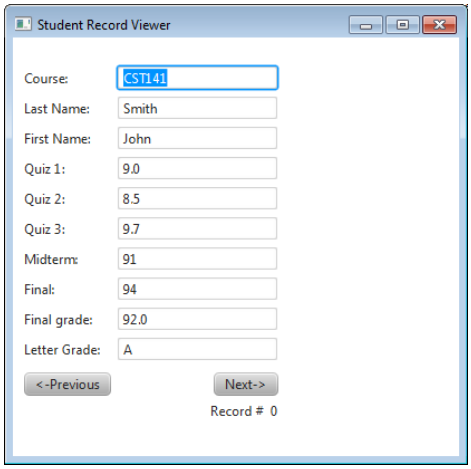
Note: Each Quiz Grade need to be multiplied by 10 so they are adjusted to a 100 pt. scale. % values can be converted to decimal by dividing by 100 so 35% is .35 in decimal.

Letter Grade can be determined using a multiway if or a switch block testing the value of the Final Grade. A switch can't test for double values so typecast them to integer such as "theIntFinalGrade = (int)theDoubleFinalGrade;" and use switch("theIntFinalGrade") as the switch statement.

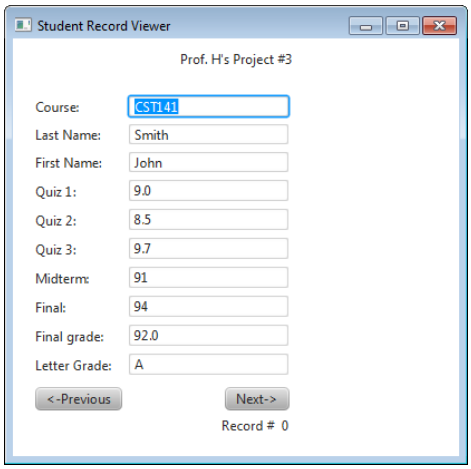
A = 90 or above, B+ = 85 – 89, B = 80 – 84, C+ = 75 – 79, C = 70 – 74, D+ = 65 – 69, D = 60 – 64, F = and all other grade values.

You can't set the value of a TextField to double so you must convert the calculated Final Grade to String such as "tfFinalGrade.setText(String.valueOf(finalGrade));".

Confirm that the changes work before moving on (see example on next page).

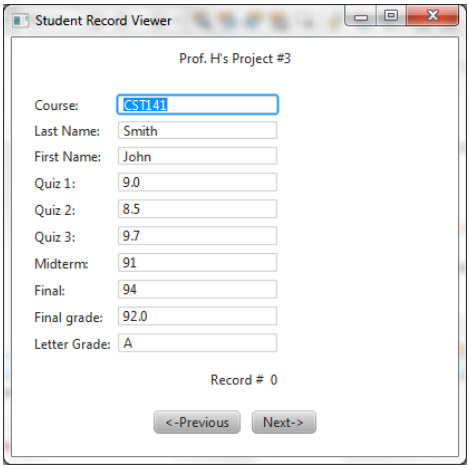


Step 3: Define a BorderPane called mainBorderPane. Add a label called mainTitle with the text "Your Last Name's Project #3" to the Top region of mainBorderPane with the setTop() method. Set the padding to 10 for top, right, bottom, and left. Center the text with "BorderPane.setAlignment(mainTitle,Pos.CENTER);". Add the centerPane, after all of the fields and buttons are added to it, to the center of mainBorderPane with the setCenter() method. See page 556 – 558 for information on BorderPanes. Set the mainScene to mainBorderPane (currently it is set to centerPane). Confirm that the program runs before moving on.



Step 4: Create an HBox and add the previous and forward buttons to the HBox. Set the spacing between objects in the HBox to 10 and set the alignment to Center. Define the padding with setPadding() as: new Insets(top = 10, right = 10, bottom = 10, left = 10).

Add the HBox to the bottom of mainBorderPane with setBottom(). HBox and VBox are shown on pages 558 – 560.

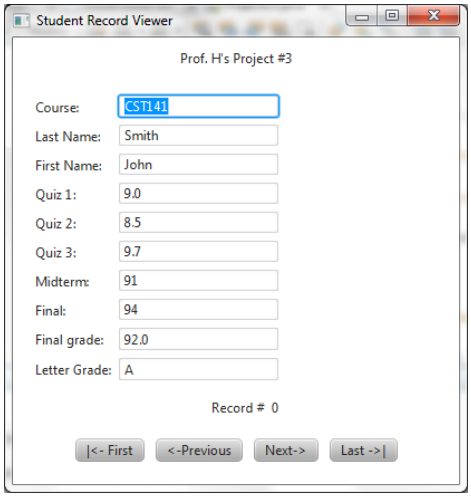


You will notice that the fields are now scrunched. This means we need to adjust the height of the stage again. Try to increase the size until the fields are no longer compressed.

Step 5: Chapter 15 discusses various ways to define an Event Handler Object to respond to button clicks and other events in a GUI environment. The author creates a separate Class that implements EventHandler, then creates an object of that Class type, and passes the object as an argument to the button object's setOnAction() method (see page 587). The author then creates an anonymous object as the argument of setOnAction() but also introduces you to creating Inner Classes (see pages 592 – 593). The author then defines Anonymous Inner Class Handlers as the argument to the setOnAction() method (see pages 594 – 597 paying attention to the top of page 595.) We are then introduced to simplifying the coding of Anonymous Inner Class Arguments to setOnAction() with Lambda Expressions starting on page 597 but you must have Java 8 installed. In any case, the setOnAction() method must pass an EventHandler object as an argument to the method. I show you another form of creating the EvenHandler object that uses a single Anonymous Inner Class Object creation for all buttons and returns the object to the invoking command. The invoking command is defineActionEvent("button text identifier") and is the argument to the setOnAction() method invocation.

Create two new buttons with the text "|<- First" and "Last ->|". These buttons are to take you to record 0 for first and to the last record with last. Add the buttons to the HBox so that the order is "First", "Previous", "Next", and "Last". Set the action with setOnAction() using the text of the argument for defineActionEvent() as "First" for the First button and "Last" for the Last button. Modify the case block in the handle() method of the anonymous object creation block of theHandlerObject variable assignment in the defineActionEvent() method. For "First" you are to set "currentRecordIndex" to 0. For "Last", you are to set "currentRecordIndex" to the last element in fileRecords with "fileRecords.size() -1". If the record index is already at 0

and the user presses the First button, you can display a message similar to the Previous button when the user has reached the first record. Do the same for the Last button as is done when the last record has been reached and the user presses the Next button.



Step 6: Change the font size of the title to make it bigger. Change the colors of the buttons so the First and Last are the same and the Next and Previous are the same (but not the default color). Change the color of the background and the color of the field labels. Change the color of the label being used as the title in the Top region of mainBorderPane. Note Mandatory: See if you can change the buttons to Red when you reach the end or beginning of the file along with printing out a message. Make sure the button goes back to the original color when another button not producing an error is pressed.

Copy your source code to the Project Assignment Template and save the file using the format described for the previous Project Assignments. Show screenshots that demonstrate the records changing for the various button presses and the output shown in the Text Console Window of NetBeans.

