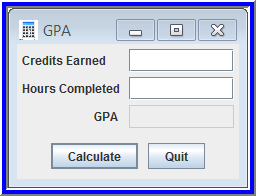
# Purpose

Practice with a simple **GUI** exercise that uses **JFrames**, **JButtons**, **JTextFields**, **JLabels, JPanels, layout managers**, and event handlers. The context is a **GUI** program to **calculate** the **GPA** based on input of **number** of **Credits** **Earned** and **number** of **Hours** **Completed**. The main window should look something like this when the application starts:

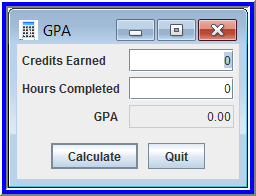
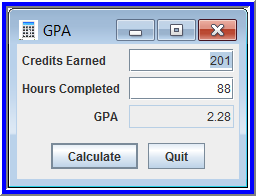


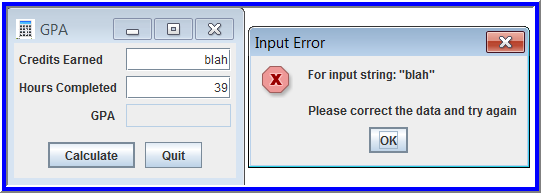
# Specifications

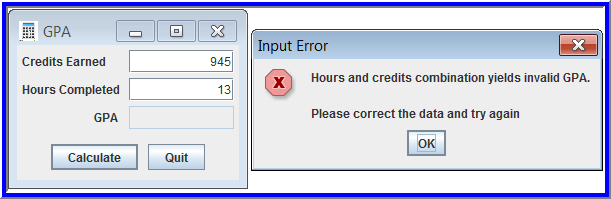
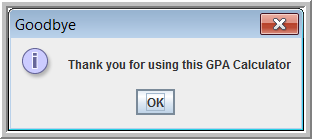
* A window size of about 240 x 180 should work reasonably well. Adjust as needed. Give a very short caption.
* All code files should be in a **package** named **gui.**
* **Comment/document** the project (files, classes, methods, and lines as needed) according to course policies
* Input for **Credits** and **Hours** should be integers. **Credits** and **Hours** ≥ **0**. If **Hours** and **Credits** are **0**, the **GPA** is **0**. If **Hours** **> 0**, then **GPA = Credits/Hours. Credits, Hours** are **int** and **GPA** is **double**, so **casting** may be needed.
* **Input errors** include
  + Negative values
  + Blank Credits or blank Hours
  + Non-integers
  + Values for which the calculated **GPA** is not between **0.00** and **4.00** (for example, input of 100 credits earned and 2 hours completed would yield an invalid **GPA** of 50.00)
* Use the exception handling mechanism to deal with input errors. Use **JOptionPane** dialogs to display **error messages**. The **GPA** field should be empty after an input error.
* When a **JTextField** has the focus, all of its current text value should be selected automatically
* Display **GPA** with two places after the decimal in a **JTextField**; **center** all windows on the screen.
* When **Calculate** is clicked, calculate and display the **GPA** based on **Credits** and **Hours** values. Display error messages if input values are invalid. Make the **Calculate** button the **default**, triggered by pressing the **Enter** key.
* When **Quit** button is clicked, display a **Goodbye** message and close the application (**System.exit** method)
* Use **private inner classes** for **event handlers**
* The **driver** should be a **separate class** as **always** with the preferred content**.**
* Use **getText** and **setText** methods of **JTextField** to retrieve/assign **String** values from/to **JTextFields**
* All **numeric input/output fields** should be **right-justified** – see screenshots.
* **Hint**: my solution uses three **panels**: **main** for the entire **content** **pane**, **upper** for **input**/**output**, **lower** for **buttons**. The **upper** **panel** uses a **grid** **layout** with 2 columns and 3 rows. The **main** **panel** uses a **border** **layout** (using only two of the regions). All **numeric input/output fields** should be **right-justified** – see screenshots.

# Screenshots of the Running Application

Some screenshots are shown here to help you visualize what your solution should be/do:



# Deliverables

You must be present in the lab and complete this in the time allotted. By the end of the class period today , submit this as **CW7 – GPA-GUI** using the conventions described in the **Course Facts** on the course web site. Be sure to submit all folders and files necessary to run the program successfully.