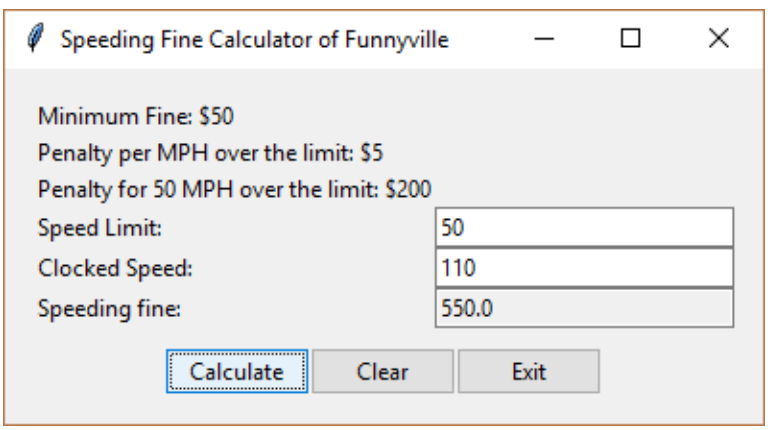
# Programming Assignment:

# GUI Application for Speeding Fine Calculator

In this assignment, you will be completing the code for a GUI application that helps user calculate the speeding fine given the penalties, the speeding limit and the clocked speed. You are given the GUI starter program (**SpeedingTicketGUI-starter.py**) and the business tier class (**SpeedingFineCalculator** in **speedingfine.py**) representing the policies of the town. You don't need to change anything in the business tier. You need to only complete the presentation tier by implementing a few methods. Here's a sample run of the application:




Specifically,

1) Download SpeedingTicketGUI-starter.py and speedingfile.py files.
2) Read the existing code that has the following implemented:
    a. **SpeedingFineFrame** class:
        i. **__init__**: The constructor calls the super class's constructor, initializes business class object and the three DoubleVars to be used with text Entry boxes.

ii. **initComponents**: Method to initialize the GUI components. This method calls initMainFrame and initButtonsFrame methods which in turn add the GUI components and set them up.

iii. **exit:** The event-handler for the exit button.

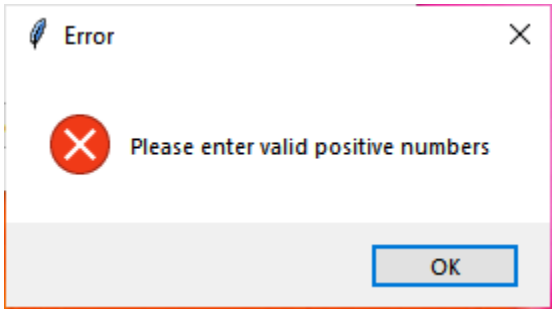
b. **main** method: That creates a root window, creates and initializes the SpeedingFindFrame class object, and starts the mainloop.

3) Implement the following methods:

a. **initMainFrame**: This creates a new Frame object and uses grid method to add it to the parent frame (self). Create and place the 6 labels and 3 text entry boxes in this frame as shown in the layout below. Make the entry box for speeding fine read-only. Connect the DoubleVar attributes declared in the constructor to the text entry boxes.

b. **initButtonsFrame**: This creates a new Frame object and uses grid method to add it to the parent frame (self). Create and place three buttons to this frame using grid method. Add the corresponding event-handlers to the buttons.

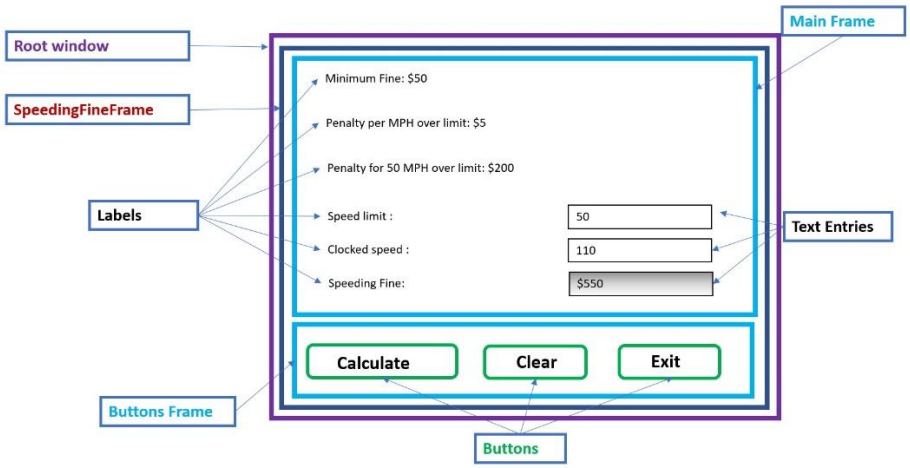
c. **calculateFine**: Implement this event-handler for the 'Calculate' button. This will read the values of Entry boxes corresponding to the speeding limit and clocked speed. Next it will call the calculateSpeedingFine method on self.speedingFineCalculator object to calculate the fine, and populate the text entry box with the fine. This method should handle the error case when User enters negative or non-numeric values in the two entry boxes and show the error message box as seen below. It should clear all the entry boxes when an error is encountered.




d. **clear:** Add the implementation of this event-handler function for the 'Clear' button. This will reset all the text entry boxes to 0.

Assume user will enter valid numbers in the text entry boxes. No need for input validation and error checking.

Here's the layout of the different GUI components in the app:




Save and submit the file as first_last_SpeedingFineGUI.py. Be sure to add the block comment at the top.