* Handle the descriptive wording in the textbox when entering text.
* This is a fun and relatively small program, you will write.
* Don’t change the design, just add event handlers and code.

Requirements

1. When no text has been entered into the textbox, display a label (with a faint foreground color), in the text box. The label wording is provided in the Tag property.
2. When the user starts entering text, the label goes away and the user’s text is the only characters in the textbox. In that case, the foreground color changes to black.
3. If the user removes all of the text, redisplay the label with the faint color, and place the cursor at the beginning of the textbox.
4. When the user clicks off of the textbox (Lost Focus event), display what’s in the textbox into the other read-only textbox.
5. If user goes back to entering text (Got Focus event), clear what’s in the read-only textbox.
6. You might want to use these textbox events:
   1. TextChanged – When the text has changed (more or less) in that control.
   2. GotFocus – Triggered when the cursor is placed in that particular control.
   3. LostFocus – Triggered when the cursor leaves that particular control.
7. You may need to know how to manipulate strings with such string methods as **.indexOf(), .Length(), substring(), ToString()**
8. I have already designed the window for you. Don’t change the design.
   1. Supplied name for the 2 textboxes, the Window, and the buttons.
   2. The window cannot be resized and gave the Window a title.
   3. Note the TAG property for the 1 textbox. It’s useful here.
   4. I already declared two brush colors.
9. Implement the EXIT button
10. Implement the RESET button. It will reset the main textbox to its label wording, and the read-only textbox to blank.

Deliverables

* Zip up the entire project folder and submit it.