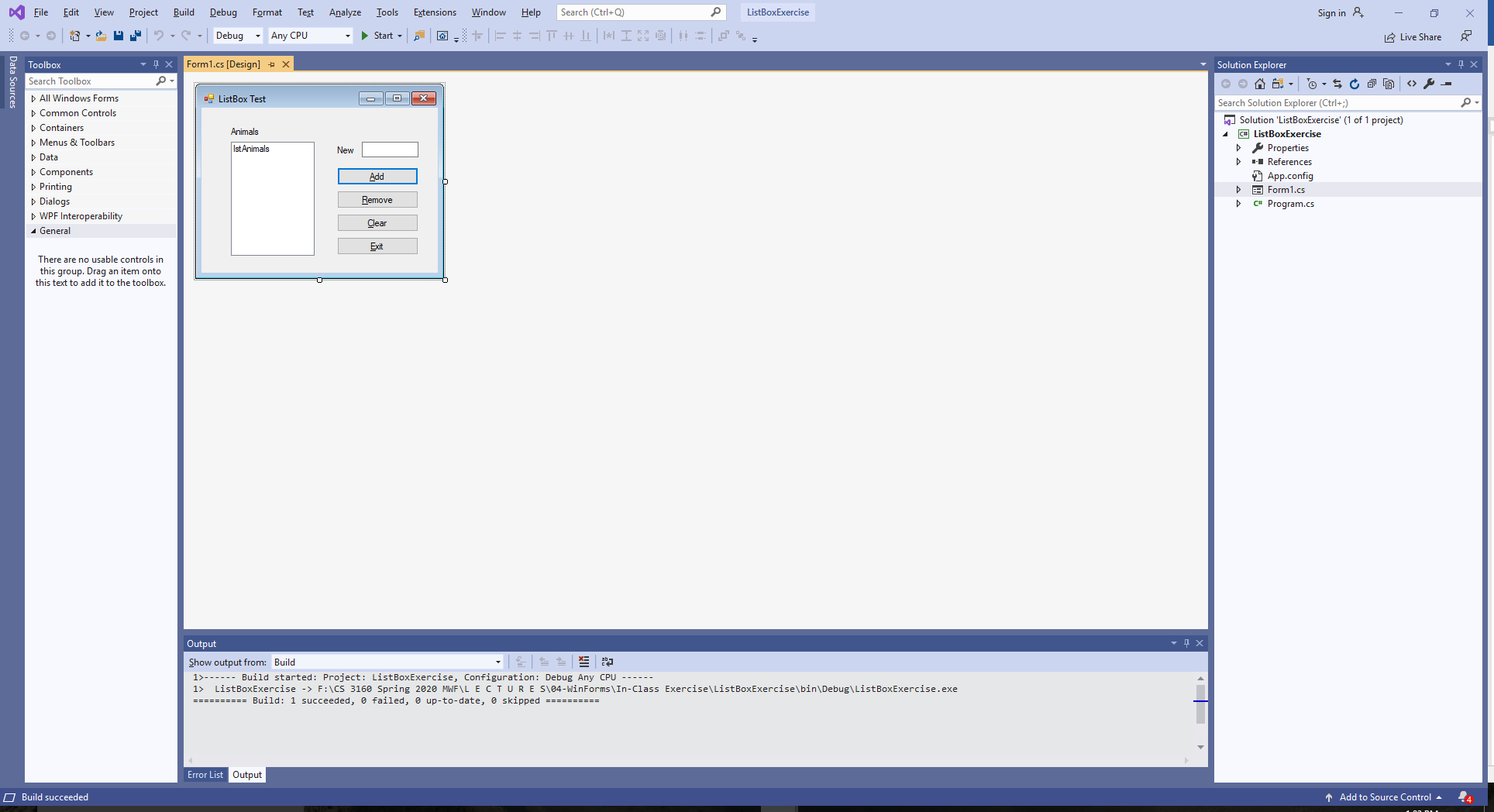
# Purpose

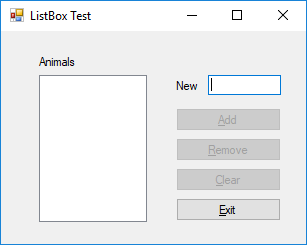
Practice creating a Windows Forms Application using Visual Studio 2019, setting properties, events, handling on click event, code-behind, and MessageBox.

# Instructions

1. Launch Visual Studio 2019, create a new C# Windows Forms Application (WinForm).
2. Pin Toolbox and select Common Controls.
3. Create a form as pictured below.



1. When the form is shown to the user initially should look like this, i.e., some controls are disabled, the focus should be in the New textbox and the only two buttons enabled should be the Add and the Exit. Utilize the Form Load event to enable/disable controls or set controls' Enabled property appropriately during design time.



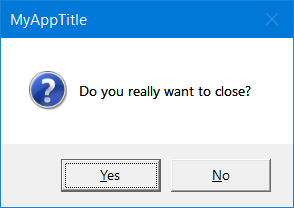
1. When the user types an animal name, enable the Add button.
2. When the user presses the Add button, add the animal to the list box. Enable the Remove buttons and the Clear button since now there is at least one item in the Listbox. Set focus back to the animal text box but clear its contents.
3. When the Clear buttons is pressed, clear all contents of the list box control.
4. When use selects an animal in the list box, enable the Remove button.
5. When the user clicks on the Remove button, remove the animal from the list. If the list is empty, disable the Remove and Clear buttons. After remove, select top items in the list box, if any are left.
6. Set form AcceptButton to be the Add button, and the CancelButton to be the Exit button.
7. Close the form when the Exit button is pressed.

# Additional Instructions ~ Bonus Lab Opportunity

You are encouraged to work in pairs. Both students must be present in class during this lab activity. In order to earn BONUS points both students should submit compressed solution folder and completed Word document via Canvas by midnight of the class activity day. Turn this sheet back to instructor during class to initiate grading activity.

Team member #1: \_\_\_\_Sidney Sanders\_\_\_\_\_\_\_ Team member #2: \_\_\_\_\_\_\_Lian Sandoval\_\_\_\_\_\_\_\_\_\_

1. For this exercise, use the animal listobox application we started last time in class. Make sure all of it is finished first before adding elements in this bonus opportunity.
2. Add a MessageBox() to ask the user if they really want to exit when the Exit button or the “X” at the top of the form is pressed. Create a on FormClosing event handler for your code.



**DialogResult r;**

**r = MessageBox.Show("Do you really want to close?");**

|  |  |
| --- | --- |
| Where did you put the code provided above? Record your answer to the right. | Code was written in the exit button function and the form1\_closing function. |
| Test your app by clicking on the Exit button and on the “X” at the top of the form. Did the MessageBox come up in both cases? Record the findings to the right. | Yes, the message box shows for both buttons clicks. |
| Modify the text box to have a more user friendly title rather than the default. How did you do it? | By adding a comma and adding a new title in quotes. |
| How did you get the ? icon to display in the message box? | When adding an icon you add another comma within the parenthesis and type MessageBoxIcon. And add the type of icon you want to show. |

1. Add validation to the textbox to ensure no digits are entered. Implement validation via the on KeyPress event handler.

**private void FilterTextBox(object sender, System.Windows.Forms.KeyPressEventArgs e)**

**{**

**// Let the system handle the key for letters**

**if (Char.IsLetter(e.KeyChar))**

**e.Handled = false;**

**else**

**// Ignore: tell the system we handled the character**

**e.Handled = true;**

**}**

|  |  |
| --- | --- |
| Test your app and try to provide numeric input. Record what happens to the right. | Numbers are not accepted; they won’t even enter the textbox when typing. |
| Copy code in the event handler to Copy/Paste butter. Now delete the event handler. Recompile. Record what happens to the right. | (confusing question…unsure what is asked, I commented out the code) Numbers appear in the text box and can be added to text box. |
| Paste it back in. Recompile. Record what happened to the right. | Numbers are not accepted. |
| Now remove the event handler by selecting properties of the textbox control and select Clear. Recompile. Record what happened to the right. | Unsure what properties to clear out, if all properties are cleared, add button doesn’t hide when supposed to. Taking out the function would let numbers be added to the list box. |

1. Remove the on KeyPress handler by deleting the handler name from the events listing. Add validation to the textbox via ErrorProvider to show small error icon by the text box if numbers are entered by dragging the EventProvider control onto your form. Add event handlers Validated and Validating.

**// get rid of the error notification ~ this will go into the Validated event handler**

**TextBox tb = (TextBox)sender;**

**errorProvider1.SetError(tb, "");**

**// some of the code below will be useful in the Validating event handler.**

**TextBox tb = (TextBox)sender;**

**if (tb.Text == "")**

**tb.Text = "0";**

**else if (!Regex.IsMatch(tb.Text, "^[a-zA-Z]+$")){**

**e.Cancel = true;**

**MyErrorProvider.SetError(tb, "Invalid animal name");**

**tb.SelectAll();**

**}**

|  |  |
| --- | --- |
| After dragging ErrorProvider onto your form, change its name to MyErrorProvider. Describe how you accomplished that to the right. | Changing the name you just need to click on the errorProvider and in the design property under name you can rename it. |
| Change blink rate of the error icon to 150. | Change can be made in the behavior property in BlinkRate |
| What namespace did you have include to use Regex class? | using System.Text.RegularExpressions; |