

Assignment 6

1. Listbox And Combobox

Adding Items Normally (via Designer)

In Visual Studio, you can add items to a `Listbox` or `Combobox` directly in the designer:

1. **Open the Designer:** Open your Windows Forms or WPF form in the designer view.
2. **Select the Control:** Click on the `Listbox` or `Combobox` you want to add items to.
3. **Access Properties:** Open the Properties window (usually on the right side).
4. **Edit Items:**
 - For a `Combobox`, find the `Items` property. Click on the ellipsis `...` button next to it. This opens the "String Collection Editor" where you can add, remove, and edit items.
 - For a `Listbox`, the procedure is similar. Find the `Items` property and click the ellipsis `...` to open the editor.
5. **Add Items:** Use the editor to add items one by one.

Adding Items Programmatically (Using a `for` Loop)

For Listbox

To add items to a `Listbox` programmatically using a `for` loop,

```
// Example for a Windows Forms application

// Assuming listBox1 is the Listbox control on your form
for (int i = 1; i <= 10; i++)
{
    listBox1.Items.Add("Item " + i);
}
```

For Combobox

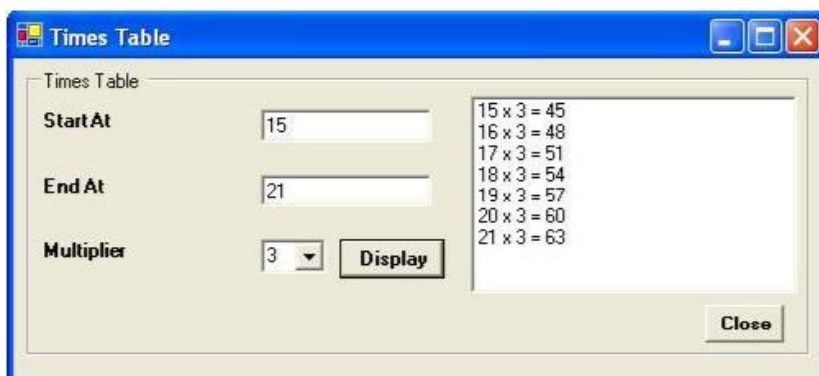
To add items to a `Combobox` programmatically using a `for` loop,

```
// Example for a Windows Forms application

// Assuming comboBox1 is the Combobox control on your form
for (int i = 1; i <= 10; i++)
{
    comboBox1.Items.Add("Item " + i);
}
```

Q1

create the following interface by giving Suitable Names for all controls.



- a. Complete the click event procedure of the **Display button** to calculate and display the times table.
(hint: use **for loop**)

Methods in a Windows Forms Application

Example 1: Method with Return Type and Method without Return Type

Let's define a method that calculates the sum of two numbers and use it in a button click event handler.

```
namespace WindowsFormsApplication1
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
        // Method with a return type
        public int AddNumbers(int a, int b)
        {
            return a + b;
        }

        // Method with no return type
        public void ShowMessage(string message)
        {
            MessageBox.Show(message);
        }
        private void btnCalculate_Click(object sender, EventArgs e)
        {
            // Example usage of AddNumbers
            int num1 = 5;
            int num2 = 3;
            int sum = AddNumbers(num1, num2);

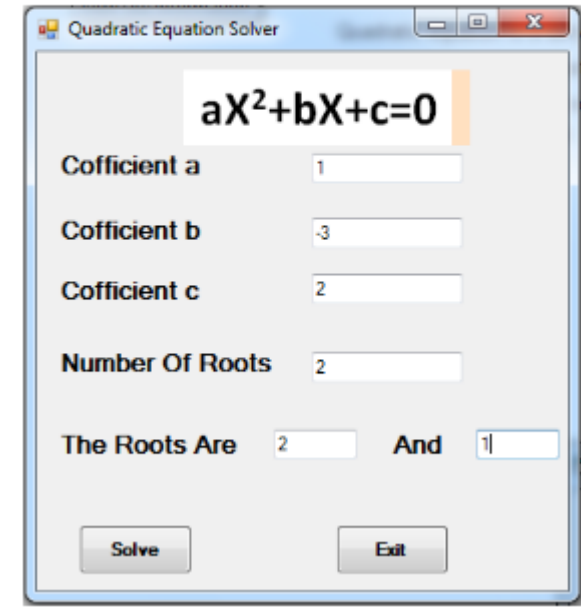
            // Example usage of ShowMessage
            ShowMessage("The sum is: " + sum);
        }

        private void btnShowMessage_Click(object sender, EventArgs e)
        {
            // Example usage of ShowMessage
            ShowMessage("Hello, this is a message!");
        }
    }
}
```

Q2

The quadratic equation solver is used to determine the number of roots the equation has as well as to compute the roots. It uses the determinant b^2-4ac to solve the problems. If $b^2-4ac>0$, then it has two roots and if $b^2-4ac=0$, then it has one root, else it has no root. To obtain the roots, the program uses the standard quadratic formula.

$$X = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$



I. Write function procedure to calculate determinant of the equation and return its value.

II. Write required code segment to Display **Number of roots** and **Roots** of the quadratic equation when you enter data and click on the **Slove** button.

Global Variables

Global Variables are variables that are accessible from anywhere in the code within the same class or namespace. In C#, the term "global variable" is commonly associated with **fields** in a class or struct, as opposed to standalone variables like in some other languages.

Global Variables (Fields)

- Defined at the class level.
- Accessible from any method within the class (instance fields) or across the class (static fields).
- Life span lasts as long as the object or application runs.

Local Variables

Local Variables are variables that are declared within a method, block, or constructor. They are only accessible within the scope they are defined.

Local Variables

- Defined within methods or blocks.
- Accessible only within the method or block where they are defined.
- Life span lasts only during the execution of the method or block.

Example

```
namespace WindowsFormsApplication1
{
    public partial class Form1 : Form
    {
        // Instance field (global to the instance of MainForm)
        private int counter = 0;

        // Static field (global to the entire application)
        private static int totalClicks = 0;
        public Form1()
        {
            InitializeComponent();
        }
        // Method with a return type

        // Method with no return type

        private void btnIncrement_Click(object sender, EventArgs e)
        {
            // Accessing instance field
            counter++;
            lblCounter.Text = "Counter: " + counter;
        }

        private void btnShowTotalClicks_Click(object sender, EventArgs e)
        {
            // Accessing static field
            totalClicks++;
            MessageBox.Show("Total Clicks: " + totalClicks);
        }
    }
}
```

Assigning Numeric Value as a Textbox Value

```
int number = 123;  
string numberAsString = number.ToString();  
txtInput.Text = numberAsString;
```

Converting TextBox Value to numeric Value

```
try  
{  
    // Retrieve the text from the TextBox  
    string inputText = txtdisplay.Text;  
  
    // Convert the string to an integer using int.Parse()  
    int number = int.Parse(inputText);  
  
    // Display the result in the Label  
    lblResult.Text = "The number is: " + number.ToString();  
}  
catch (FormatException)  
{  
    // Handle the case where the input is not a valid number  
    lblResult.Text = "Invalid input! Please enter a valid integer.";  
}  
catch (OverflowException)  
{  
    // Handle the case where the input is too large or too small  
    lblResult.Text = "The number is too large or too small.";  
}
```

Q3

Create the following interface and write the code to do calculator tasks.

