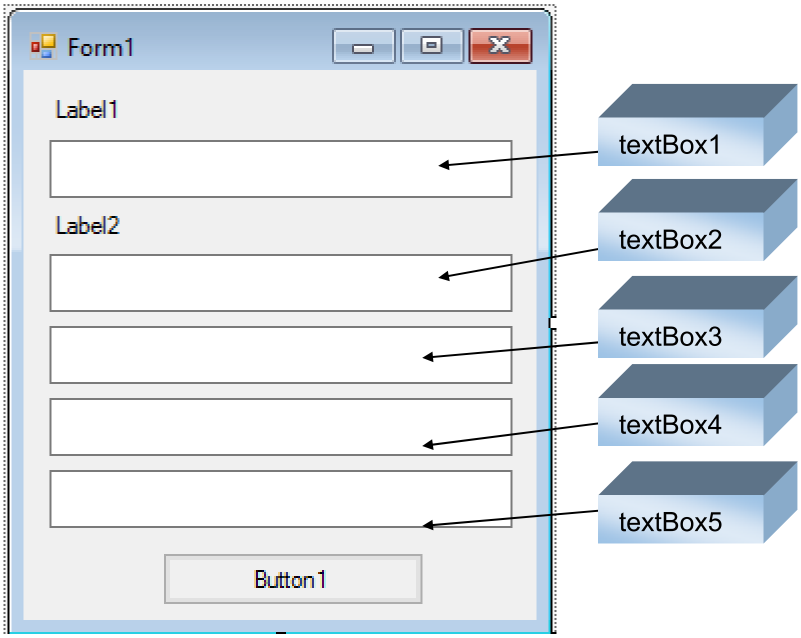
Project 4 – State Capitals

Project Design

In this project, we build a fun game for home and school. You will be given the name of a state in the United States and four possible choices for its capital city. You click on the guess of your choice to see if you are right. (We apologize to our foreign readers – perhaps you can modify this project to build a similar multiple choice type game). Click on **Next State** for another question. This project is saved as **StateCapitals** in the project folder (**\BeginVCS\BVCS Projects**).

Place Controls on Form

Start a new project in Visual C#. Place two label controls, five text boxes and two buttons on the form. When done, you form should resemble this:



Set Control Properties

Set the control properties using the properties window:

Form1 Form:

Property Name	**Property Value**
Text	State Capitals
FormBorderStyle	FixedSingle
StartPosition	CenterScreen

label1 Label:

Property Name	**Property Value**
Name	lblHeadState
Text	State:
Font Size	14
Font Style	Italic

textBox1 Text Box:

Property Name	**Property Value**
Name	txtState
BackColor	White
Font Size	14
ReadOnly	True
TextAlign	Center
TabStop	False

label2 Label:

Property Name	**Property Value**
Name	lblHeadCapital
Text	Capital:
Font Size	14
Font Style	Italic

textBox2 Text Box:

Property Name	Property Value
Name	txtCapital0
BackColor	White
Font Size	14
ReadOnly	True
TextAlign	Center
TabStop	False

textBox3 Text Box:

Property Name	Property Value
Name	txtCapital1
BackColor	White
Font Size	14
ReadOnly	True
TextAlign	Center
TabStop	False

textBox4 Text Box:

Property Name	Property Value
Name	txtCapital2
BackColor	White
Font Size	14
ReadOnly	True
TextAlign	Center
TabStop	False

textBox5 Text Box:

Property Name	Property Value
Name	txtCapita3
BackColor	White
Font Size	14
ReadOnly	True
TextAlign	Center
TabStop	False

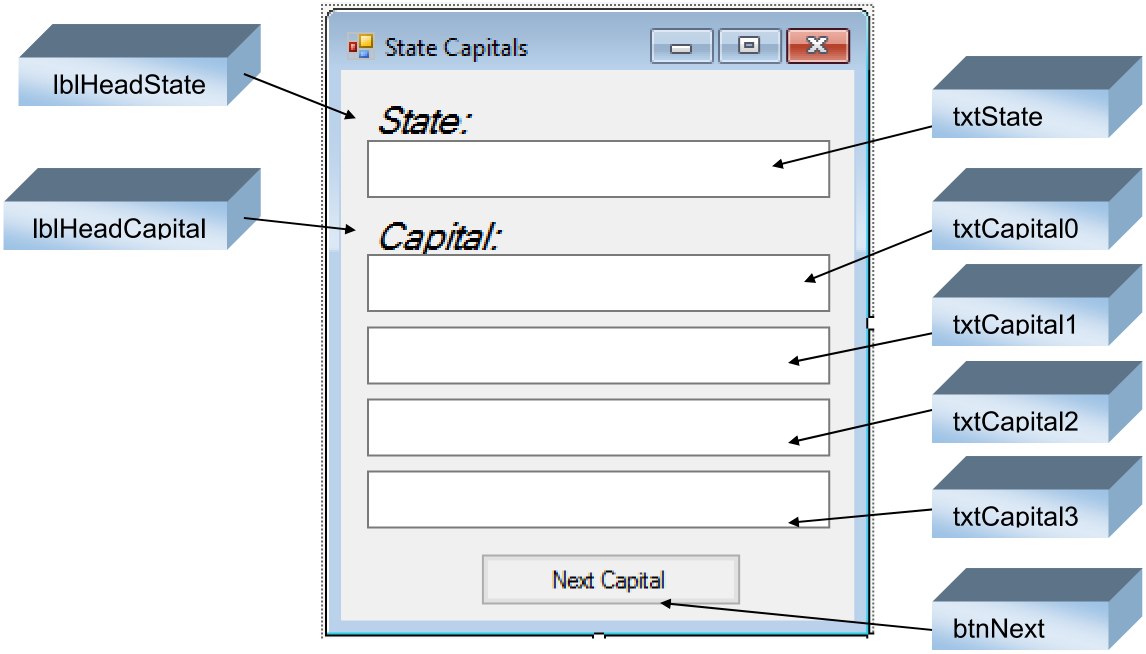
button1 Button:

Property Name	Property Value
Name	btnNext
Text	Next State

button2 Button:

Property Name	Property Value
Name	btnCheck

Resize the form so **btnCheck** does not appear (we'll use this as a general method button). When done, the form looks like this :



Write Event methods

To display a state and possible capitals, click **Next State**. Click on your choice for answer.

Put this code in the general declarations area:

```
int answer;
string[] state = new string[50];
string[] capital = new string[50];
TextBox[] listedCapital = new TextBox[4];
int capitalClicked;
Random myRandom = new Random();
```

Add this code to the **Form1_Load** event (yes, it's a lot of typing or just copy and paste from these notes):

```
private void Form1_Load(object sender, EventArgs e)
{
    // load state and capital arrays
    state[0] = "Alabama"; capital[0] = "Montgomery";
    state[1] = "Alaska"; capital[1] = "Juneau";
    state[2] = "Arizona"; capital[2] = "Phoenix";
    state[3] = "Arkansas"; capital[3] = "Little Rock";
    state[4] = "California"; capital[4] = "Sacramento";
    state[5] = "Colorado"; capital[5] = "Denver";
    state[6] = "Connecticut"; capital[6] = "Hartford";
    state[7] = "Delaware"; capital[7] = "Dover";
    state[8] = "Florida"; capital[8] = "Tallahassee";
    state[9] = "Georgia"; capital[9] = "Atlanta";
    state[10] = "Hawaii"; capital[10] = "Honolulu";
    state[11] = "Idaho"; capital[11] = "Boise";
    state[12] = "Illinois"; capital[12] = "Springfield";
    state[13] = "Indiana"; capital[13] = "Indianapolis";
    state[14] = "Iowa"; capital[14] = "Des Moines";
    state[15] = "Kansas"; capital[15] = "Topeka";
    state[16] = "Kentucky"; capital[16] = "Frankfort";
    state[17] = "Louisiana"; capital[17] = "Baton Rouge";
    state[18] = "Maine"; capital[18] = "Augusta";
    state[19] = "Maryland"; capital[19] = "Annapolis";
    state[20] = "Massachusetts"; capital[20] = "Boston";
    state[21] = "Michigan"; capital[21] = "Lansing";
    state[22] = "Minnesota"; capital[22] = "Saint Paul";
    state[23] = "Mississippi"; capital[23] = "Jackson";
    state[24] = "Missouri"; capital[24] = "Jefferson City";
    state[25] = "Montana"; capital[25] = "Helena";
    state[26] = "Nebraska"; capital[26] = "Lincoln";
    state[27] = "Nevada"; capital[27] = "Carson City";
    state[28] = "New Hampshire"; capital[28] = "Concord";
    state[29] = "New Jersey"; capital[29] = "Trenton";
    state[30] = "New Mexico"; capital[30] = "Santa Fe";
    state[31] = "New York"; capital[31] = "Albany";
    state[32] = "North Carolina"; capital[32] = "Raleigh";
    state[33] = "North Dakota"; capital[33] = "Bismarck";
    state[34] = "Ohio"; capital[34] = "Columbus";
    state[35] = "Oklahoma"; capital[35] = "Oklahoma City";
    state[36] = "Oregon"; capital[36] = "Salem";
    state[37] = "Pennsylvania"; capital[37] = "Harrisburg";
    state[38] = "Rhode Island"; capital[38] = "Providence";
    state[39] = "South Carolina"; capital[39] = "Columbia";
    state[40] = "South Dakota"; capital[40] = "Pierre";
    state[41] = "Tennessee"; capital[41] = "Nashville";
    state[42] = "Texas"; capital[42] = "Austin";
    state[43] = "Utah"; capital[43] = "Salt Lake City";
    state[44] = "Vermont"; capital[44] = "Montpelier";
    state[45] = "Virginia"; capital[45] = "Richmond";
    state[46] = "Washington"; capital[46] = "Olympia";
    state[47] = "West Virginia"; capital[47] = "Charleston";
    state[48] = "Wisconsin"; capital[48] = "Madison";
    state[49] = "Wyoming"; capital[49] = "Cheyenne";
    // Set listed capital labels
    listedCapital[0] = txtCapital0;
    listedCapital[1] = txtCapital1;
    listedCapital[2] = txtCapital2;
    listedCapital[3] = txtCapital3;
    // set first question
    btnNext.PerformClick();
}
```

The **btnNext_Click** event method generates the next multiple choice question:

```
private void btnNext_Click(object sender, EventArgs e)
{
    int[] vUsed = new int[50];
    int[] index = new int[4];
    int j;
    // Generate the next question at random
    btnNext.Enabled = false;
    answer = myRandom.Next(50);
    // Display selected state
    txtState.Text = state[answer];
    // Vused array is used to see which state capitals have
    // been selected as possible answers
    for (int i = 0; i < 50; i++)
    {
        vUsed[i] = 0;
    }
    // Pick four different state indices (J) at random
    // These are used to set up multiple choice answers
    // Stored in the Index array
    for (int i = 0; i < 4; i++)
    {
        // Find index not used yet and not the answer
        //Find value not used yet and not the answer
        do
        {
            j = myRandom.Next(50);
        }
        while (vUsed[j] != 0 || j == answer);
        vUsed[j] = 1;
        index[i] = j;
    }
    // Now replace one index (at random) with correct answer
    index[myRandom.Next(4)] = answer;
    // Display multiple choice answers in text boxes
    for (int i = 0; i < 4; i++)
    {
        listedCapital[i].Text = capital[index[i]];
    }
}
```

A new concept in this routine is the **do** loop (shaded line) to pick the different possible answers. Let's explain how this particular loop works.

The form used in this code is:

```
do
{

      [C# code]
}
while condition;
```

In this "loop," the C# code between the **do** line and the **while** line is repeated as long as the specified **condition** is true. See if you can see how this loop allows us to pick four distinct capital cities for the multiple choice answers (no repeated values).

The event methods for clicking the capital city text box controls simply identify which text box was clicked and “clicks” the hidden **btnCheck** button:

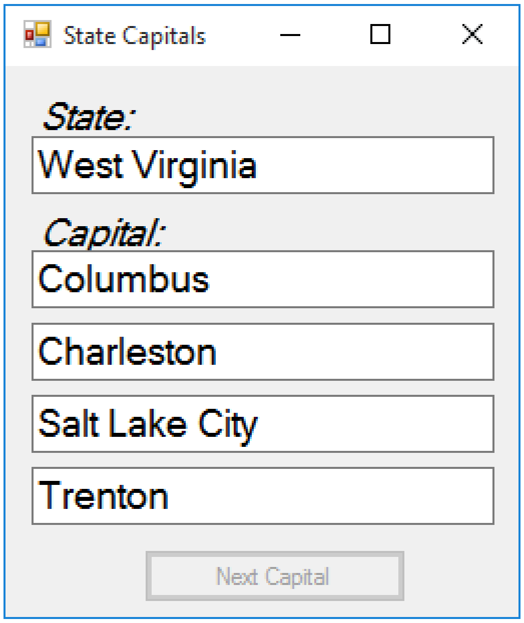
```
private void txtCapital0_Click(object sender, EventArgs e)
{
    capitalClicked = 0;
    btnCheck.PerformClick();
}

private void txtCapital1_Click(object sender, EventArgs e)
{
    capitalClicked = 1;
    btnCheck.PerformClick();
}

private void txtCapital2_Click(object sender, EventArgs e)
{
    capitalClicked = 2;
    btnCheck.PerformClick();
}

private void txtCapital3_Click(object sender, EventArgs e)
{
    capitalClicked = 3;
    btnCheck.PerformClick();
}
```

We write a **btnCheck_Click** “hidden” general method to check the answer (when it is selected by clicking a text box control):

```
private void btnCheck_Click(object sender, EventArgs e)
{
    // If already answered, ignore click
    if (listedCapital[capitalClicked].Text != "" &&
btnNext.Enabled == false)
    {
        // Check clicked answer
        if (listedCapital[capitalClicked].Text ==
capital[answer])
        {
            // Correct answer - clear out other answers and
enable Next button
            for (int i = 0; i < 4; i++)
            {
                if (i != capitalClicked)
                {
                    listedCapital[i].Text = "";
                }
            }
            btnNext.Enabled = true;
            btnNext.Focus();
        }
        else
        {
            // Incorrect answer - clear out selected answer
            listedCapital[capitalClicked].Text = "";
        }
    }
}
```

Run the Project

Save your work. Run the project. A state name and four possible capital cities will be displayed. (Study the code used to choose and sort the possible answers – this kind of code is very useful.) Choose an answer. If correct, the other answers will be cleared. Click **Next State** to continue. If incorrect, your choice will be cleared. Keeping answering until correct. Here's a run I made where I missed on my first guess:



State:
West Virginia

Capital:
Columbus

Charleston

Salt Lake City

Trenton

Other Things to Try

This would be a fun project to modify. How about changing it to display a capital city with four states as the multiple choices? Allow the user to type in the answer (use a text box) instead of picking from a list. Add some kind of scoring system.

This program could also be used to build general multiple choice tests from any two lists. You could do language translations (given a word in English, choose the corresponding word in Spanish), given a book, choose the author, or given an invention, name the inventor. Use your imagination.