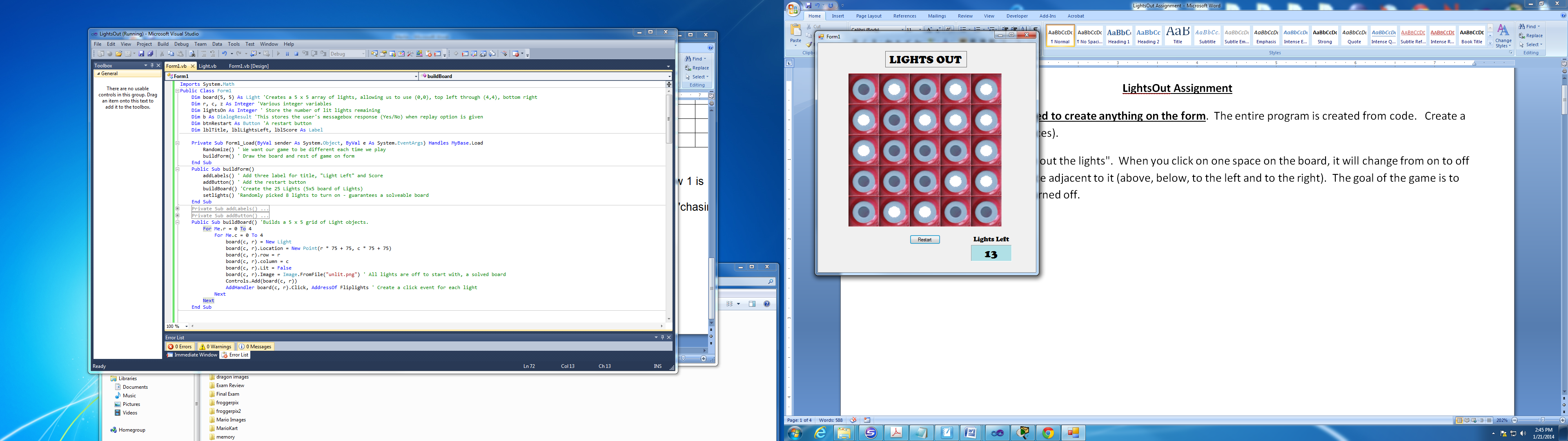
**LightsOut Assignment**

In this program, **you will not need to create anything on the form**. The entire program is created from code. Create a project called *LightsOut* (no spaces).

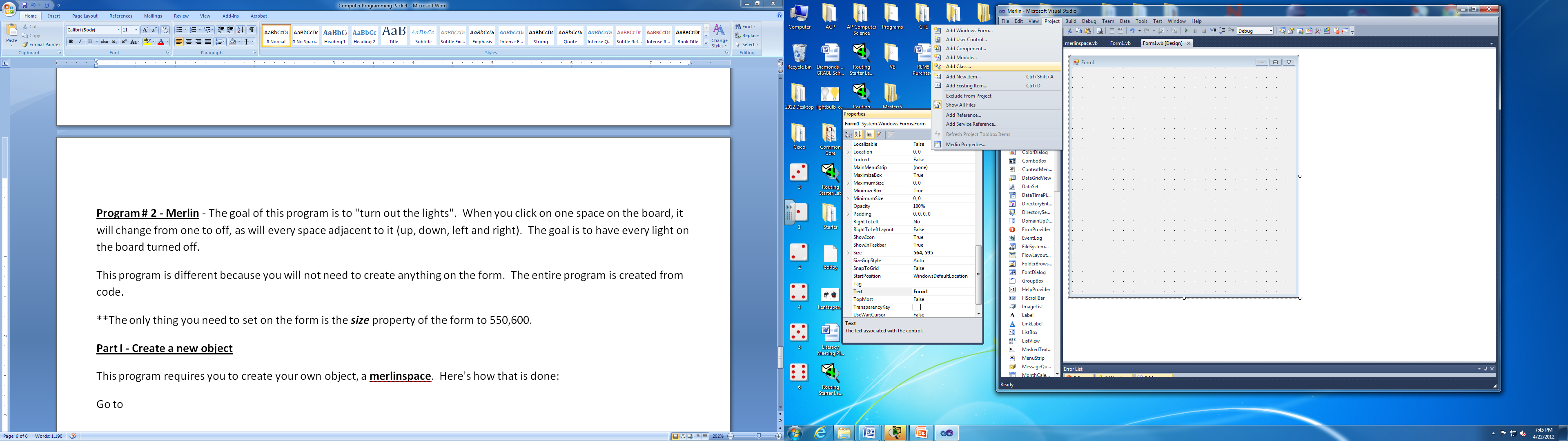
The goal of this game is to "turn out the lights". When you click on one space on the board, it will change from on to off (or vice versa), as will every space adjacent to it (above, below, to the left and to the right). The goal of the game is to have every light on the board turned off.

This program does require two images to be loaded in the debug folder. They are found in the Computer Programming/Images folder on the shared drive. They are called lit.png and unlit.png.

\*\*\*The only thing you need to set on the form design is the ***size*** property of the form to 550,600.

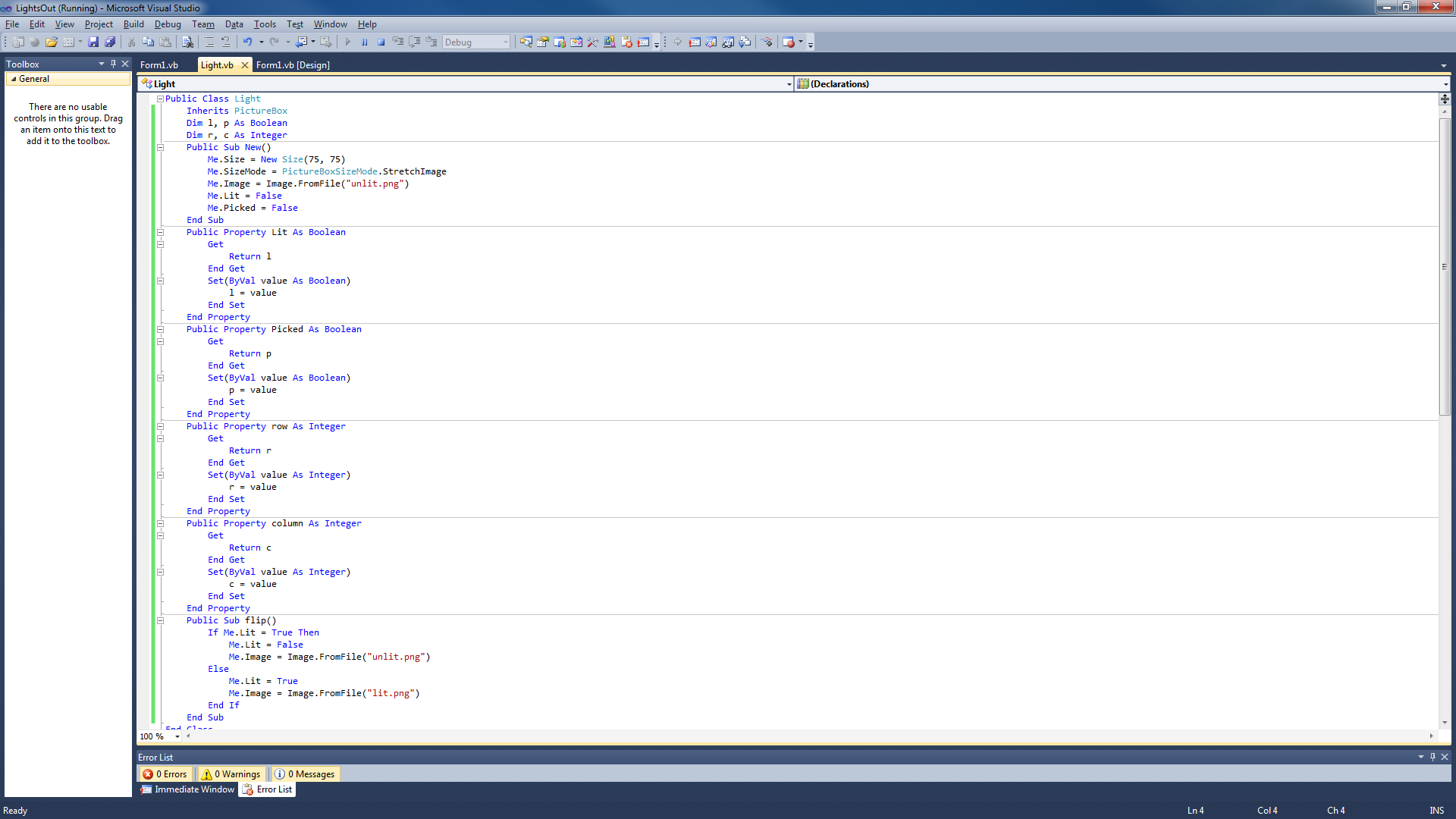
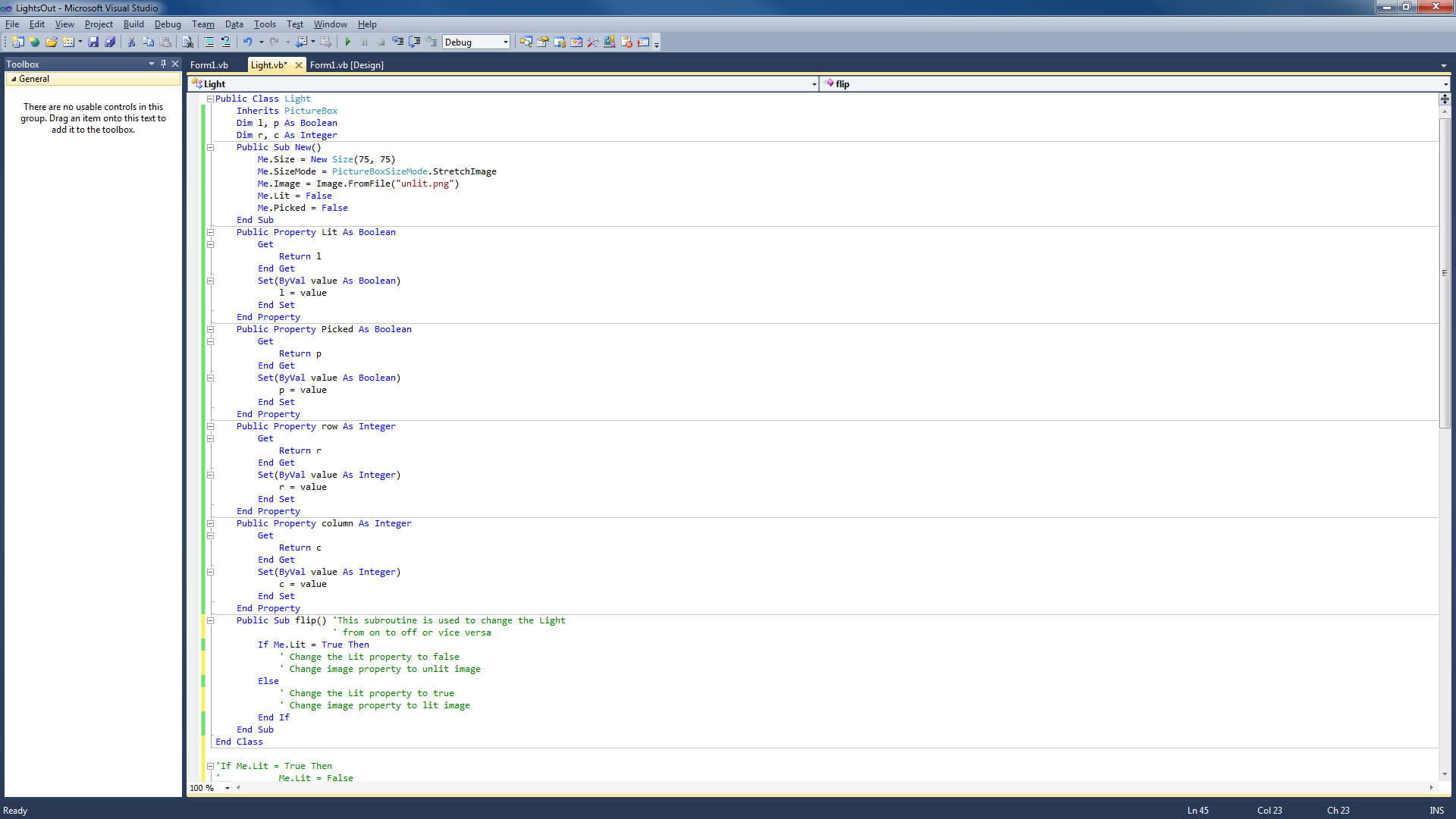
**Part I - Create a new object**

This program requires you to create your own object, a **Light**. The LightsOut board will be composed of 25 Light objects, arranged in 5 columns and 5 rows.

Go to your Project menu and select Add new Class. (See below)

Call your new class ***Light*** (not class1). Click Add and a coding page will appear.

The following code will create your new Light object and set up four additional properties that are specific to your Light object. You will also create a flip subroutine which will be used to flip your Light object from "on" to "off" or vice versa.



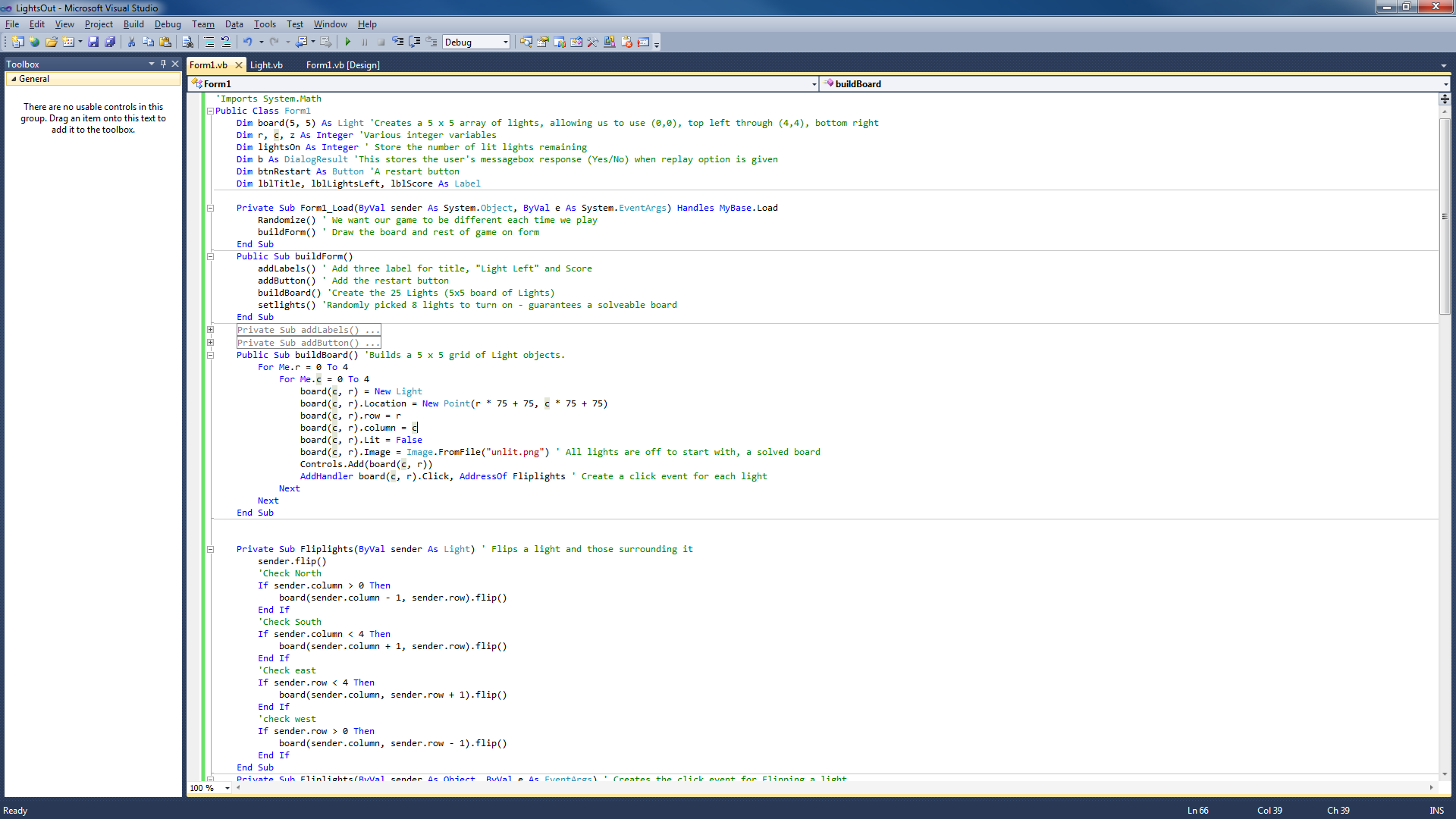
You have now created a new object called a *Light*. It is essentially a picturebox, but we can also now record a row and column coordinate within it. We also have a ***lit*** property to determine if it is lit or unlit. Finally we have a picked property which is used to draw and randomize the game board.

We also have a flip subroutine which you must finish writing the code for. It is used to simply flip the light from "on" to "off" or vice versa.

**Part II. Form1 Variables**

**Part III. Form1 Events**

The only event you will create is a Form1\_Load, which will be created by double clicking on the form. There will be several other subroutines, which you will create from scratch. The order of the subroutines is not important. Just make sure you end each of the subroutines with an *End Sub* line. Below you find Form1\_Load and *buildForm* called from it.



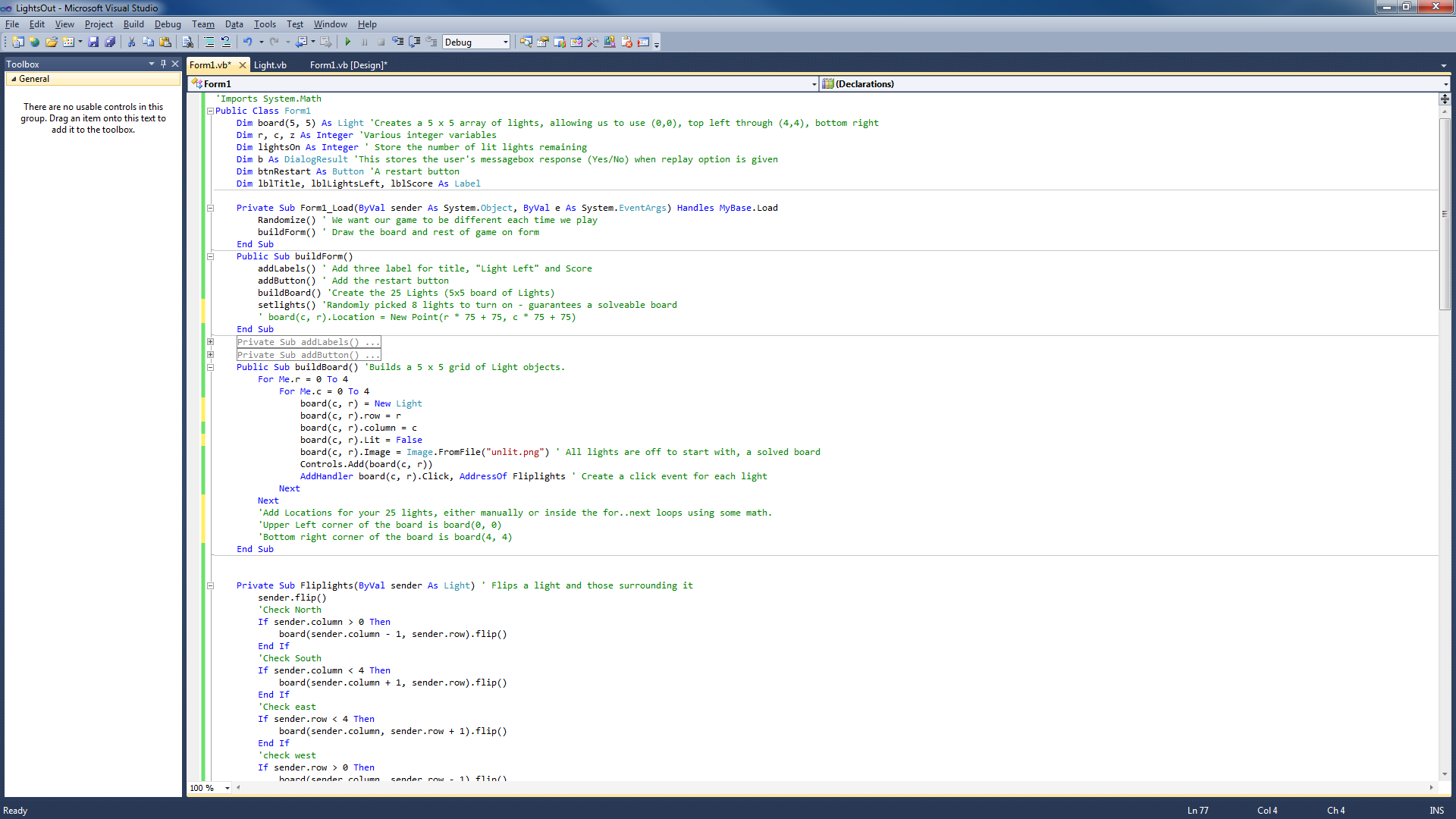
**addLabels Subroutine**

You will create this subroutine yourself. In it, you must create the three labels shown on page 1. (Lights Out Title, Lights Left, and Score (showing 13 in sample form) ) Match yours as closely to mine as possible for full credit. I used Cooper Black font, size 20 on title and score, size 12 on the "Lights Left" label.

**addButton Subroutine**

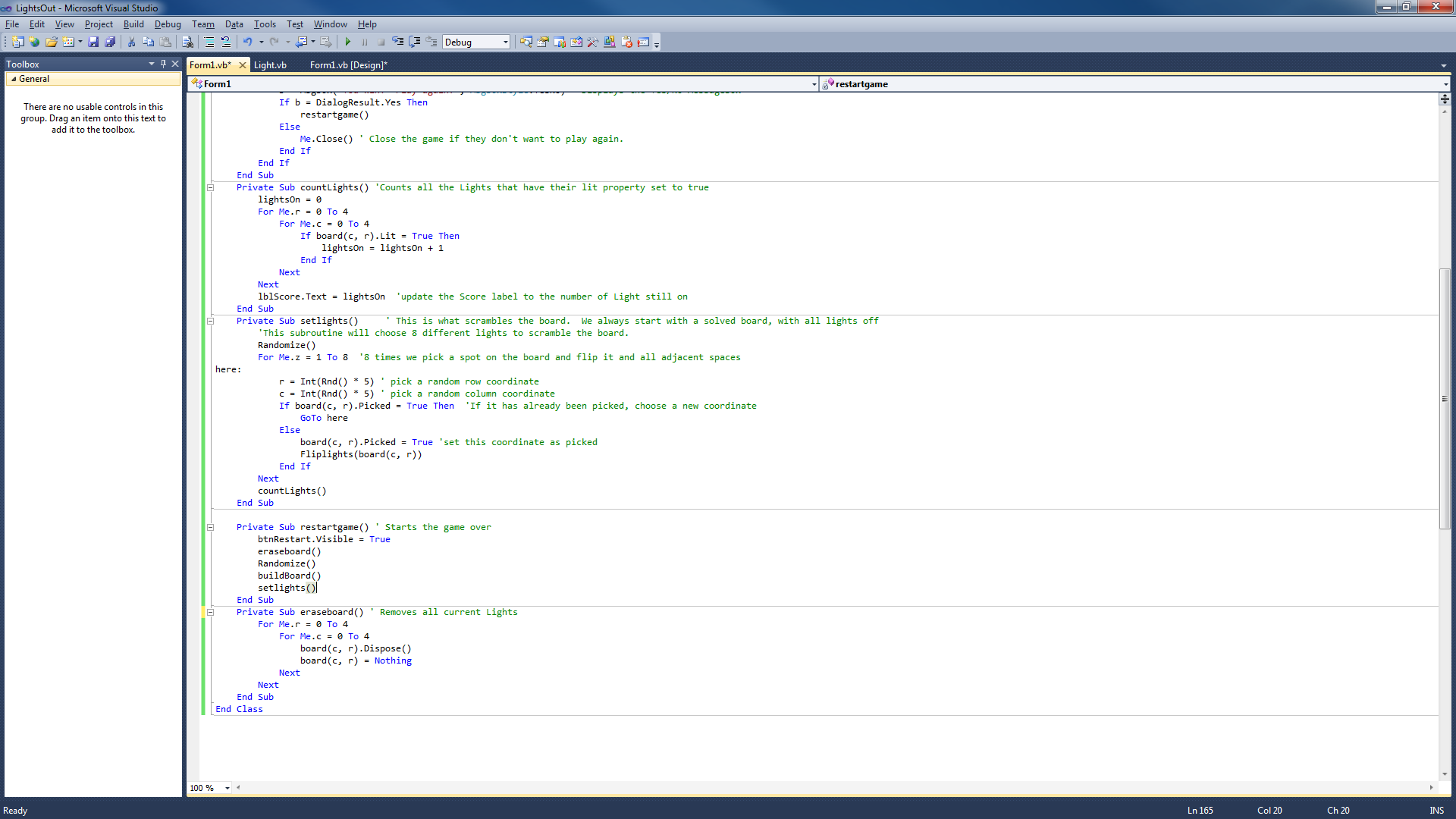
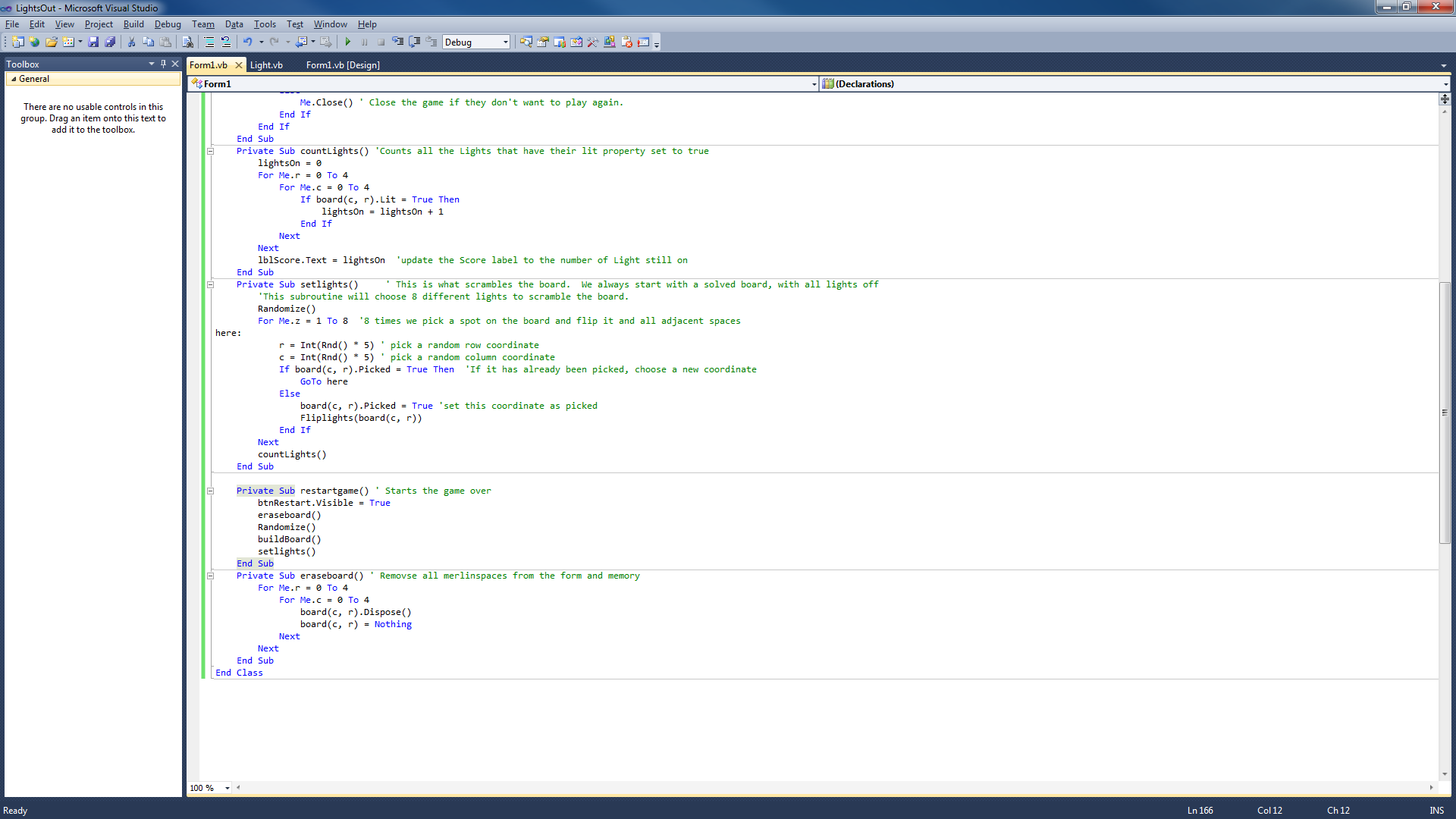
You will create this subroutine also. In it, you must create the restart button that appears just below the game board. I expect the button to be balanced in the middle of the form, just below the game board. You must also create a click event for this button using the *AddHandler* command. It should be used to call the *restartGame* subroutine. The button should be visible until the first Light is clicked upon and then it disappears. It will remain invisible until the player wins, at which time, it will reappear.

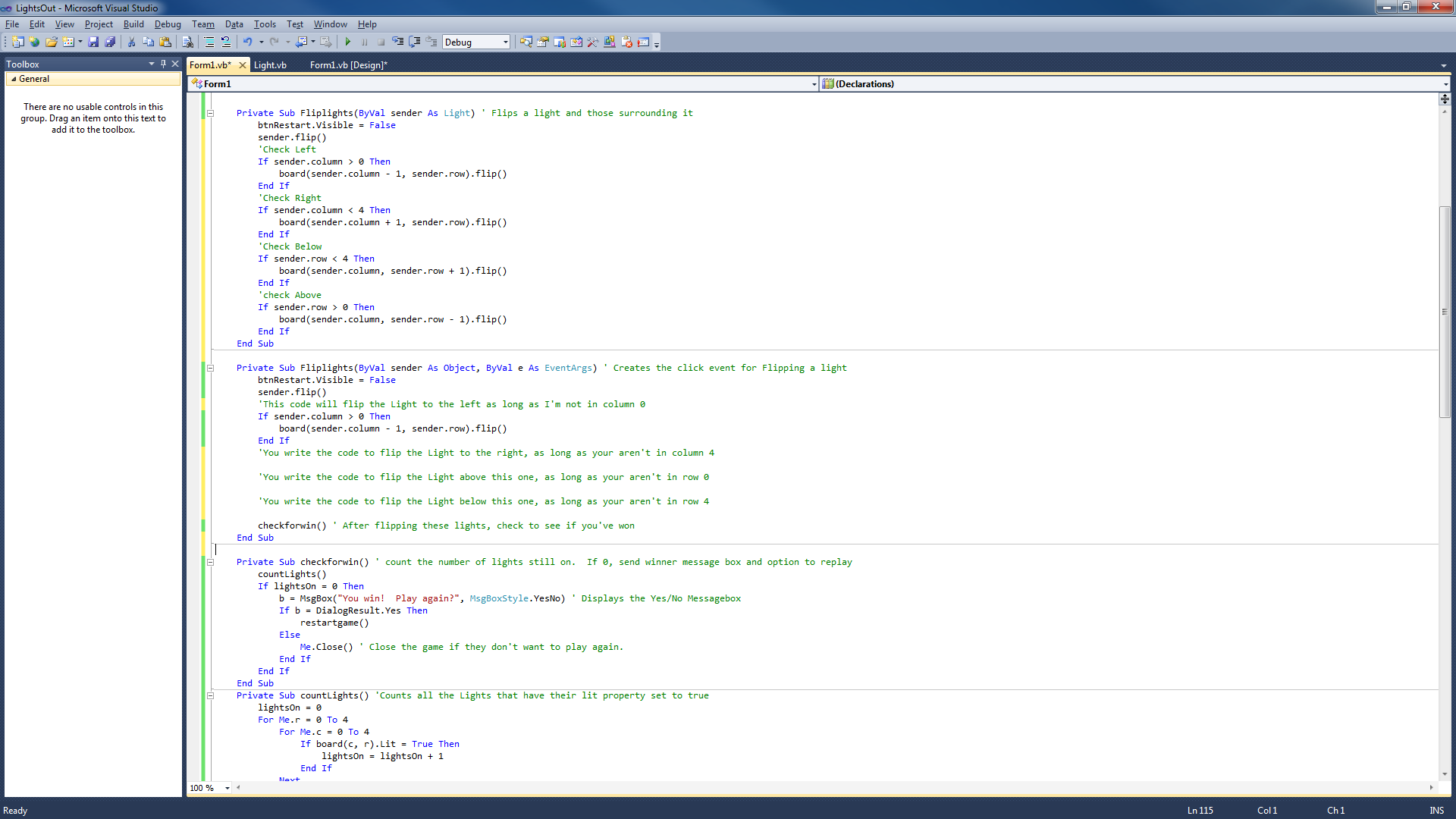
**Building the Board**



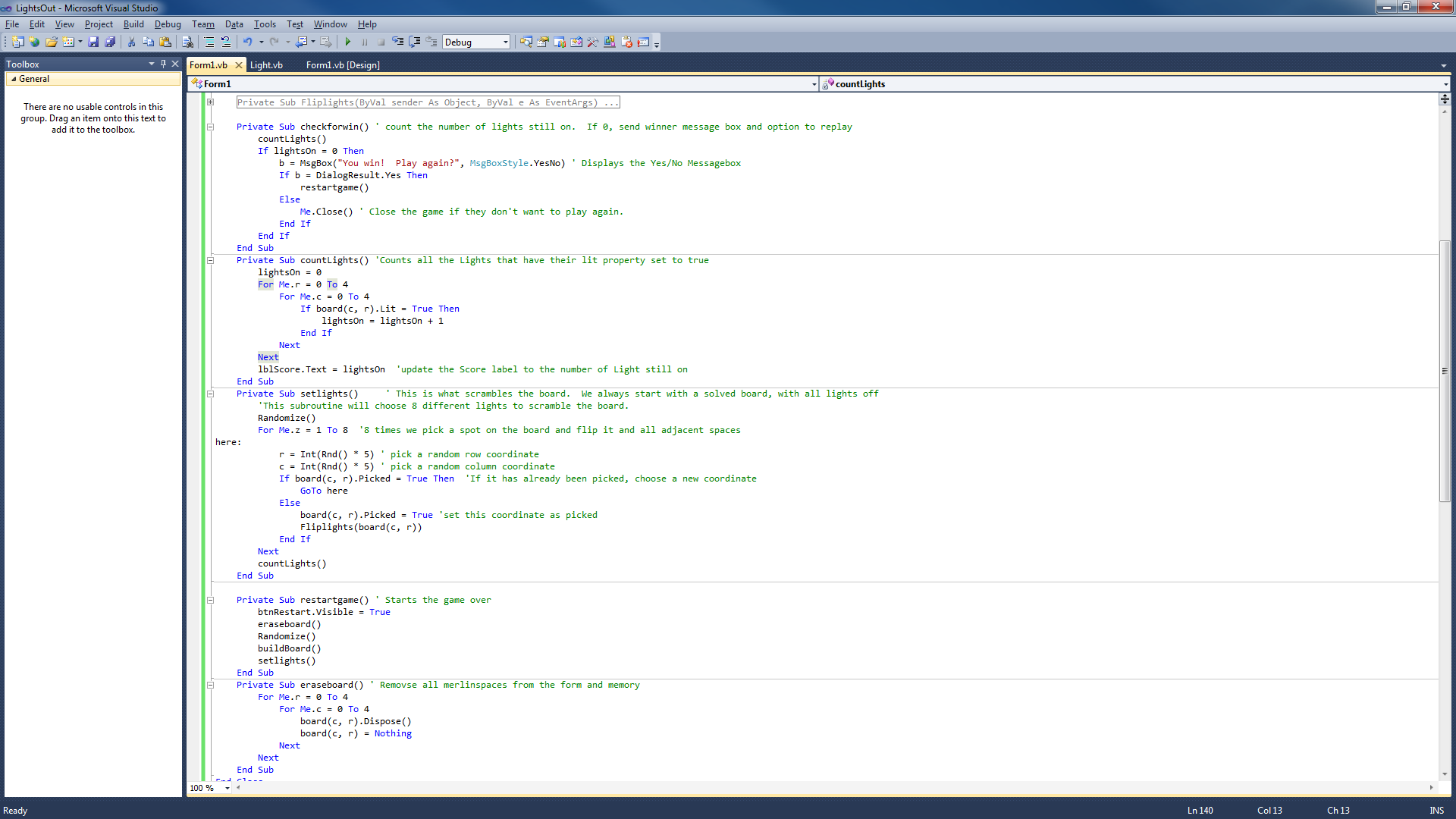
Board coordinates are expressed with the row number appearing first, then the column. The table below illustrates this:

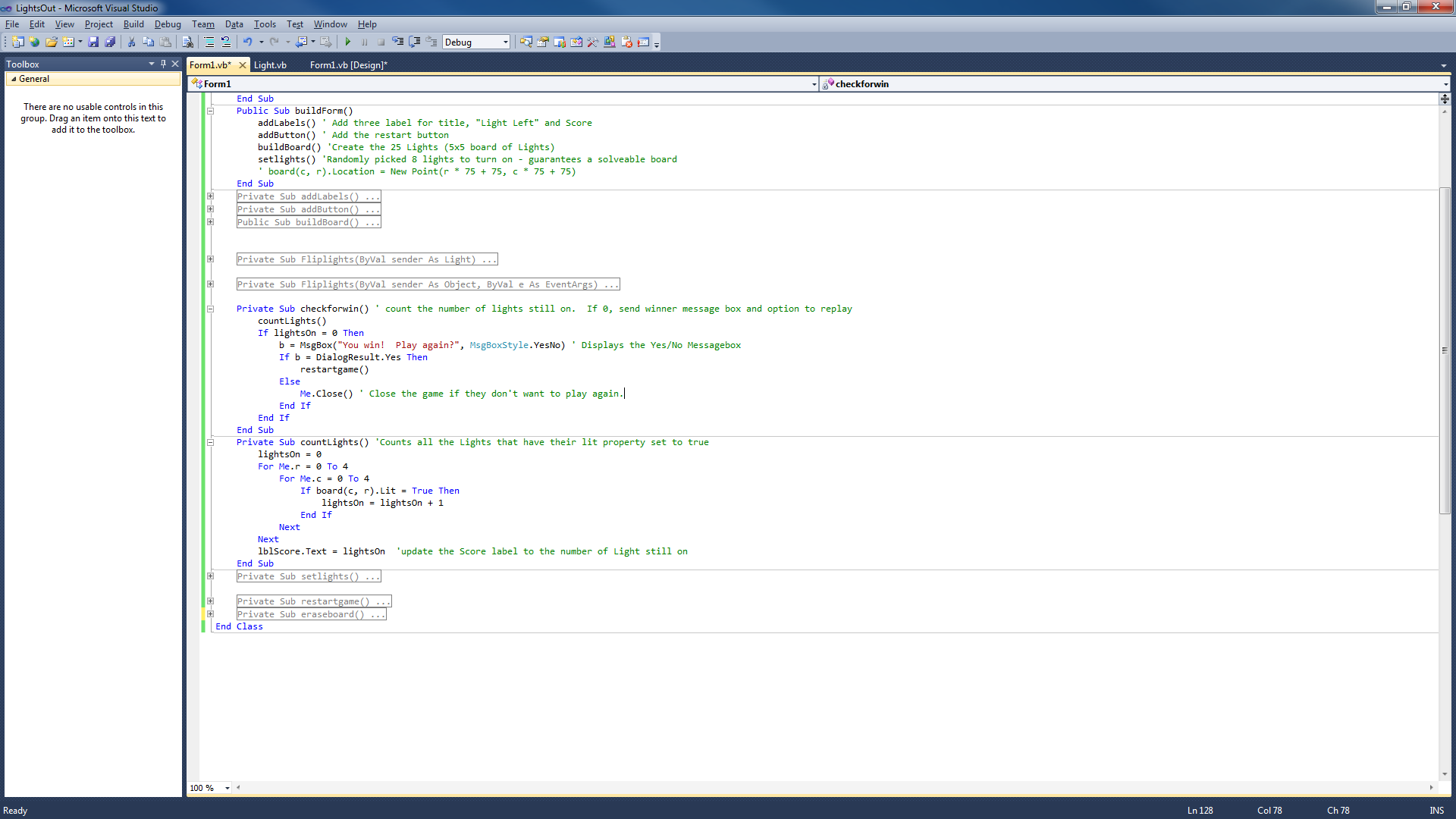
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| (0, 0) | (0, 1) | (0, 2) | (0, 3) | (0, 4) |
| (1, 0) | (1, 1) | (1, 2) | (1, 3) | (1, 4) |
| (2, 0) | (2, 1) | (2, 2) | (2, 3) | (2, 4) |
| (3, 0) | (3, 1) | (3, 2) | (3, 3) | (3, 4) |
| (4, 0) | (4, 1) | (4, 2) | (4, 3) | (4, 4) |

****

****

**\*Important\*** - Once complete, copy and paste the code for this subroutine and remove the ByVal e as EventArgs on to line and the last line, Checkforwin(). You will have two FlipLights subroutines in the end, but with different parameters (Stuff in parentheses) so it is okay.



****

**Lights Out Rubric**

|  |  |  |
| --- | --- | --- |
| **Criteria** | **Points Possible** | **Points Earned** |
| Do 25 Lights appear when you start your program? (1/4 pt each) | 6 |  |
| Do the 25 Lights appear in 5 columns and rows, perfectly lined up? | 10 |  |
| Do the three labels appear and look similar to those on page 1? (2.5 pts each) | 7.5 |  |
| Do all Lights "flip" when they are clicked upon? Test each. (1/4 pt each) | 6 |  |
| Do all adjacent spaces also flip? (1/2 pt each) | 12.5 |  |
| Does the Score label accurately output the number of lights still on? | 2 |  |
| Does the game detect a win? | 3 |  |
| Will the game reset when you click "Yes" in the message box? | 3 |  |
| Total |  |  |

**\*Note** - I will assist you once you have all the code keyed in. You are responsible for the layout of the Form and the missing code from the two *FlipLights* subroutines and the Light *flip* subroutine.

**Solving LightsOut**

1. For each light on row 0, select the space beneath it in row 1 to turn the light off. This way row 0 is completely unlit.
2. Repeat step a for rows 1-3, so that now you only have lights on row 4. This is usually called *'chasing the lights*'.
3. If the light at (4, 0) is on, click on (0, 3) and (0, 4).  
   If the light at (4, 1) is on, press (0, 1) and (0, 4).  
   If the light at (4, 2) is on, press (0, 3).
4. Repeat steps a-b, chasing the lights down and it will be magically solved.