**Validate users by using If . . . Then**

1. Start Visual Studio, and create a new Windows Forms Application project named My User Validation. The new project is created, and a blank form opens in the Designer.

2. Click the form, and then set the form’s Text property to “User Validation.”

3. Use the Label control to create a label on your form, and use the Properties window to set the Text property to “Enter Your Social Security Number.”

4. Use the Button control to create a button on your form, and set the button’s Text property to “Sign In.”

5. Click the MaskedTextBox control on the Common Controls tab in the Toolbox, and then create a masked text box object on your form below the label. The MaskedTextBox control is similar to the TextBox control that you have been using, but by using MaskedTextBox, you can control the format of the information entered by the user into your program. You control the format by setting the Mask property; you can use a predefined format supplied by the control or choose your own format. You’ll use the MaskedTextBox control in this program to require that users enter a Social Security number in the standard nine-digit format used by the U.S. Internal Revenue Service.

6. With the MaskedTextBox1 object selected, click the Mask property in the Properties window, and then click the ellipses button in the second column. The Input Mask dialog box opens, showing a list of your predefined formatting patterns, or masks.

7. Click Social Security Number in the list. Although you won’t use it now, take a moment to note the option, which you can use later to create your own input masks using numbers and placeholder characters such as a hyphen (-).

8. Click OK to accept Social Security Number as your input mask. Visual Studio displays your input mask in the MaskedTextBox1 object.

9. Double-click the Sign In button. The Button1\_Click event procedure appears in the Code Editor.

10. Type the following program statements in the event procedure:

If MaskedTextBox1.Text = "555-55-1212" Then

MsgBox("Welcome to the system!")

Else MsgBox("I don't recognize this number")

End If

This simple If . . . Then decision structure checks the value of the MaskedTextBox1 object’s Text property, and if it equals “555-55-1212,” the structure displays the message “Welcome to the system!” If the number entered by the user is some other value, the structure displays the message “I don’t recognize this number.” The beauty in this program, however, is how the MaskedTextBox1 object automatically filters input to ensure that it is in the correct format.

11. Click the Save All button on the Standard toolbar to save your changes.

12. Click the Start Debugging button on the Standard toolbar. The program runs in the IDE. The form prompts the user to enter a Social Security number (SSN) in the appropriate format, and displays underlines and hyphens to offer the user a hint of the format required.

13. Type abcd to test the input mask. Visual Basic prevents the letters from being displayed because letters do not fit the requested format. A nine-digit SSN is required.

14. Type 1234567890 to test the input mask. Visual Basic displays the number 123-45-6789 in the masked text box, ignoring the 10th digit that you typed. Again, Visual Basic has forced the user’s input into the proper format.

15. Click the Sign In button. Visual Basic displays the message “I don’t recognize this number” because the SSN does not match the number the If . . . Then decision structure is looking for.

16. Click OK, delete the SSN from the masked text box, enter 555-55-1212 as the number, and then click Sign In again. This time the decision structure recognizes the number and displays a welcome message. Your code has prevented an unauthorized user from using the program, and you’ve learned a useful skill related to controlling input from the user.

17. Exit the program. Get Ticked off