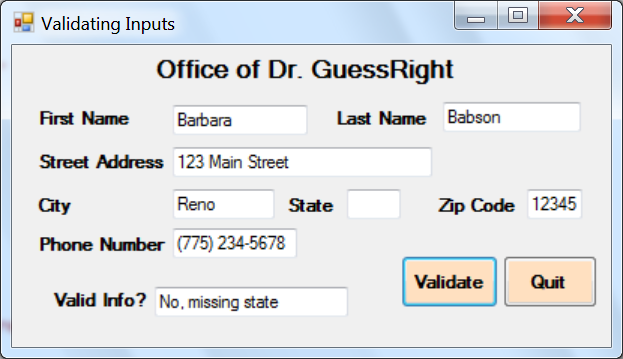
IS 350—Prof. Mark G. Simkin The Office of Dr. GuessRight



**Objective:** The objective of this assignment is to practice creating subroutines that Visual Basic applications can use to validate typical business inputs.

**Description:** The office of Dr. GuessRight asks new patients to use the input screen shown above to input their name, address, and telephone number. But patients often omit items or makes mistakes in the information they do provide. The office manager has asked you to create a user interface with validation routines that can address these problems.

You proceed as follows. To validate the data, you decide to create three subroutines. The first subroutine tests for blank textboxes for first name, last name, street address, city, and state. There are no inputs to this subroutine, but two outputs: (1) a Boolean true/false value indicating whether or not there were any problems, and (2) if there were, the reason for it (see example above).

The second subroutine tests for a valid zip code and the third subroutine tests for a valid phone number. Assume that a valid zip code must be exactly five numeric digits, and that a valid phone number uses the format “(123) 456-7890.” Both of these subroutines recognize that there can be two possible problems—either a value is missing (i.e., the user leaves it blank) or a value is not in the correct format (e.g., the user enters the four-digit zip code3 “1234”). The system should provide alternate explanations—i.e., should provide a message that says “No, missing zip code” or a message that says “No, bad zip code.”

When the user hits the “validate” button, the code for its click event should call each of these subroutines in turn to validate the data. If all data pass the test, the system should display a simple “Yes” in the final textbox of the application.

Hints: (1) For the name and address data, it is ok if the system only finds one error, even if there are several. This allows the system to display a simple error message. (2) You will find it helpful to use the Like operator for the zip code and telephone number validations. (3) The code for this assignment is extensive, but mostly repetitive. (4) If you think about it, you will realize that there are even better ways to code this problem than the three-subroutine solution suggested above.