Select only **one** problem to receive credit for this homework. Of course, you can work on both problems if you wish.

3-1: DateEntered and DateModified field on the customer form.

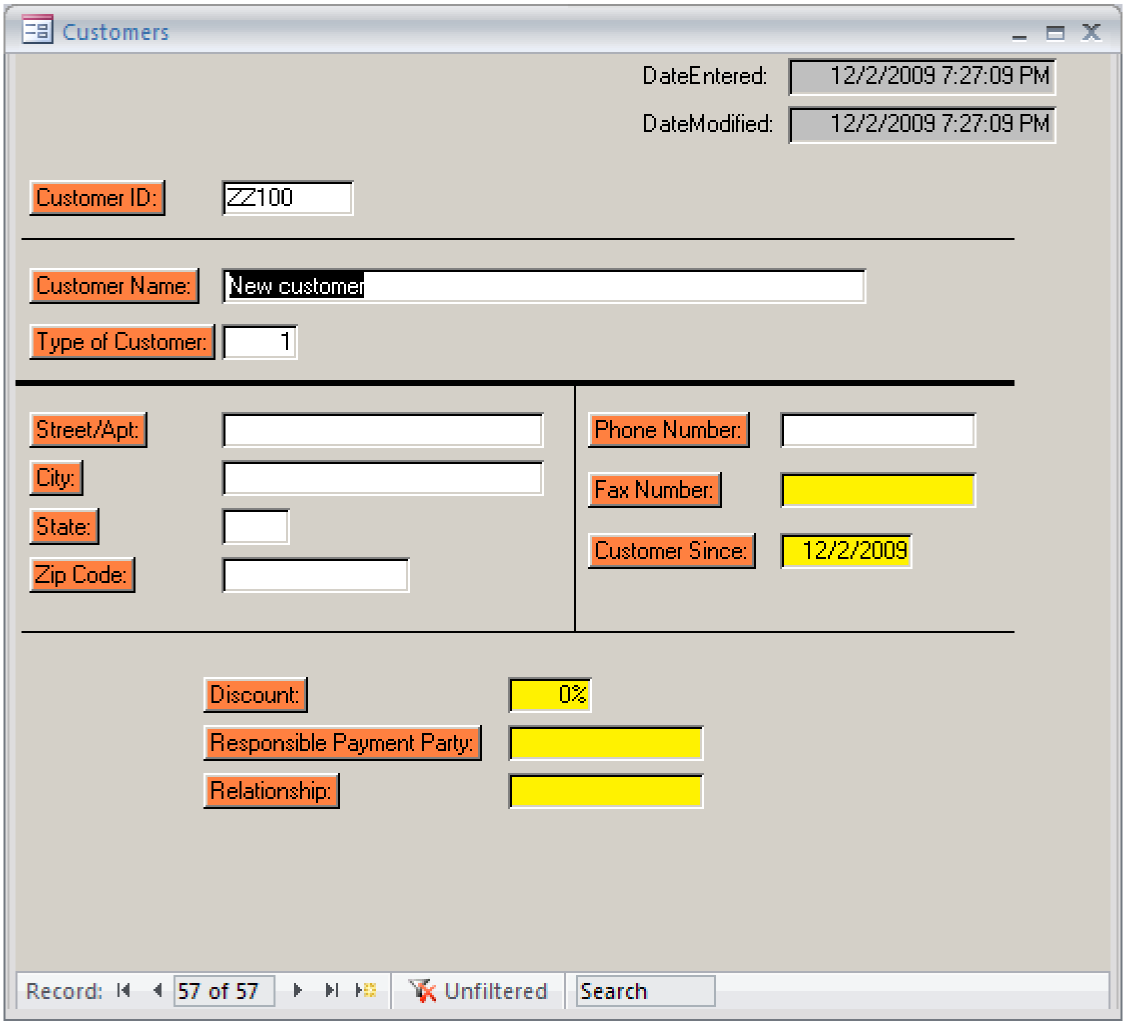
a) Go to frm_Customer in design view and save it as frm_homework_3. Navigate to the property sheet for the form, tab to Other, and set the Has Module property to No. Confirm this by clicking on Yes. This will effectively remove all VBA code from this form. Delete all the buttons in the header section.

b) Add DateEntered and DateModified fields in the header section of the customer form. Make sure to bind these fields to the database. Set the Locked property to Yes, and the Tab Stop property to No. Change the background to dark grey.

c) Create an event procedure for the Before Update event at the form level to update these two date fields.

d) If this is an existing record, change only the DateModified field to the current date/time by using the Now()function, if this is a new record, change both fields to the current date/time.
Hint: One way is to test the DateEntered field, if it is null, then it is a new record. Another option is to use the form property Me.NewRecord. If it is true, then it is a new record.

e) Test the form by modifying an existing record and adding a new record. This is required for the homework!



DateEntered: 12/2/2009 7:27:09 PM
DateModified: 12/2/2009 7:27:09 PM

Customer ID: ZZ100

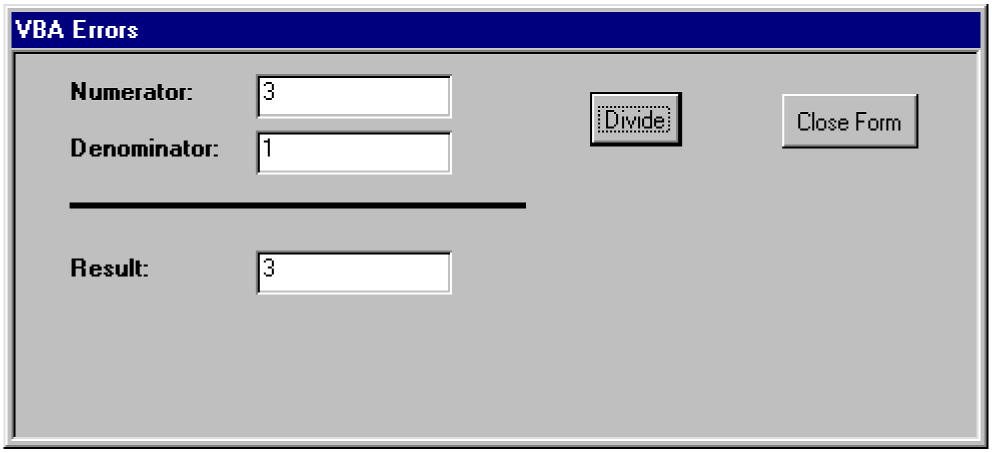
Customer Name: New customer
Type of Customer: 1

Street/Apt:
City:
State:
Zip Code:

Phone Number:
Fax Number:
Customer Since: 12/2/2009

Discount: 0%
Responsible Payment Party:
Relationship:

3-2: Create the following form including Error handling routines



VBA Errors

Numerator: 3

Denominator: 1

Result: 3

1) Create an unbound form and save it as frm_Divide as shown above.
2) Add three unbound text boxes as seen above. Use the prefix for lbl for labels and txt for text boxes. Name the controls by using the prefix and the label caption of each text box. For example: lblNumerator, txtNumerator.
3) Add a command button to close the form. Name the button cmdClose.
4) Add a command button to divide the Numerator/Denominator. Name the button cmdDivide. Use the Wizard and select any operation (for example to close a form), simply to create the error handling routine.
5) Modify the On Click event property for the cmdDivide button as follows:
   a) Delete the code pertaining to the operation chosen by the wizard (for example if you had chosen Close Form).
   b) Build an expression to perform the division.
   c) Modify the message box statement in the Error handler to include the error number and description. Use the VBA Err object.
   d) Now test the form. Generate two errors: First, divide by zero. Take a note of the error number you see in the message box statement. Second, enter a non numeric entry. Take a note of the error number.
   e) Enhance you error handler to test for these two errors to allow for a customized message. Make sure to include an Else statement at the end to handle any other errors.