

Resistor Sizing Program

In this chapter, you will learn how to use the following Visual Basic Application functions for AutoCAD to World Class standards:

- **Opening the Visual Basic Editor in AutoCAD**
- **Beginning a New Visual Basic Application**
- **Laying Out a User Input Form in Visual Basic**
- **Insert a Label into a Form**
- **Insert a Textbox into a Form**
- **Inserting More Labels and Textboxes into a Form**
- **Insert a Label into a Form to Post an Output**
- **Creating and Inserting an Image into a Form in Visual Basic**
- **Insert Command Buttons into a Form**
- **Adding a Copyright Statement to a Form**
- **Adding Comments in Visual Basic to Communicate the Copyright**
- **Programming the Compute Command Button**
- **Resetting the Data**
- **Exiting the Program**
- **Running the Program**

Open the Visual Basic Editor in AutoCAD

Opening the Visual Basic Editor in AutoCAD is essential to creating a program to automate the drawing process. In this version of the World Class CAD – Visual Basic Applications for AutoCAD, we are using AutoCAD 2009, but we just finished using all the programs in this text with a group of programmers using AutoCAD 2002. Their drawings were automatically made just as efficiently as if they were using the most recent version of the Autodesk software.

Select Tools on the Menu bar, then Macro and then Visual Basic Editor. For quick launching of the editor, press Alt – F11.



Figure A.1 – Launching the Visual Basic Editor

The Visual Basic Editor will appear on the computer desktop as a new program application. Looking down on the computer’s Task bar, we can see the AutoCAD and Microsoft Visual Basic Editor program tabs. Just single click either program tab to switch between the two applications. However, if we close the AutoCAD drawing, unlike a stand-alone version of Visual Basic, the Visual Basic Editor will close also.

For those individuals with previous Visual Basic experience, the Visual Basic Editor in AutoCAD has the same layout. The menu bar contains tools for our use as well as the four toolbars shown in Figure A.3, which are Standard, Debug, Edit and Userform. Presently, only the Standard toolbar is showing. On the left side of the workspace is the Project menu, which shows the files pertaining to this project. Below the Project menu is the Properties pane. If we remember the Properties tool in AutoCAD, using this device will be simple.

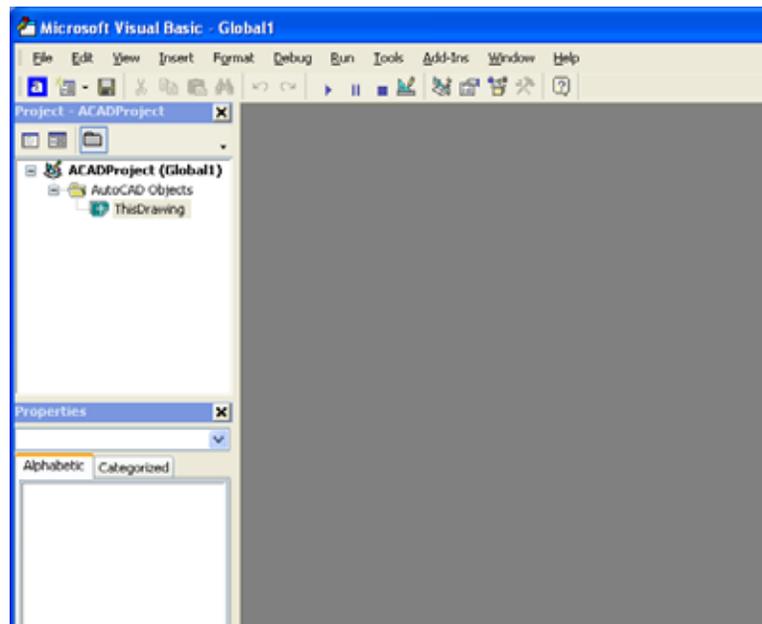


Figure A.2 – The Visual Basic Editor

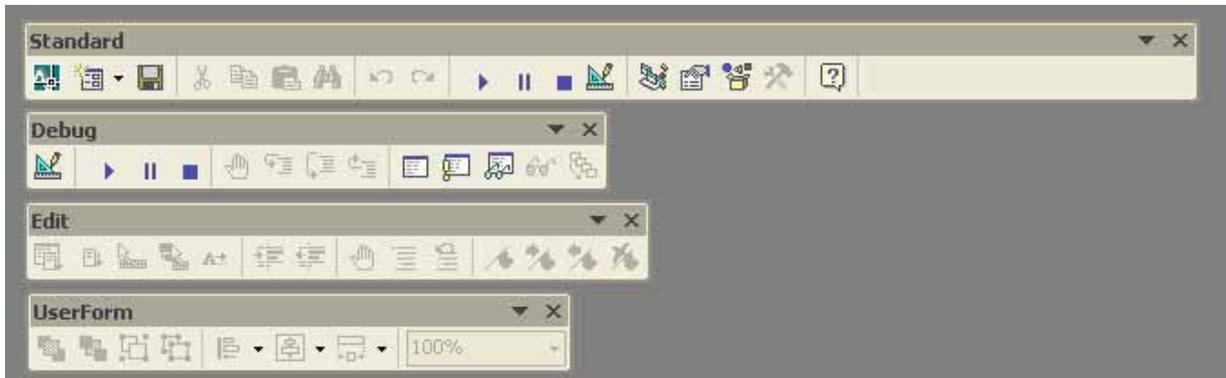


Figure A.3 – Toolbars in the Visual Basic Editor

With the Visual Basic Editor open, select **File** on the Menu Bar and select **Save Project**. We will create a folder on either the desktop or the My Documents folder called “VBA Programs”. After creating the folder, name the file “Resistor Sizing”. The file extension *dvb* as can be easily seen means DCL and Visual Basic programs.

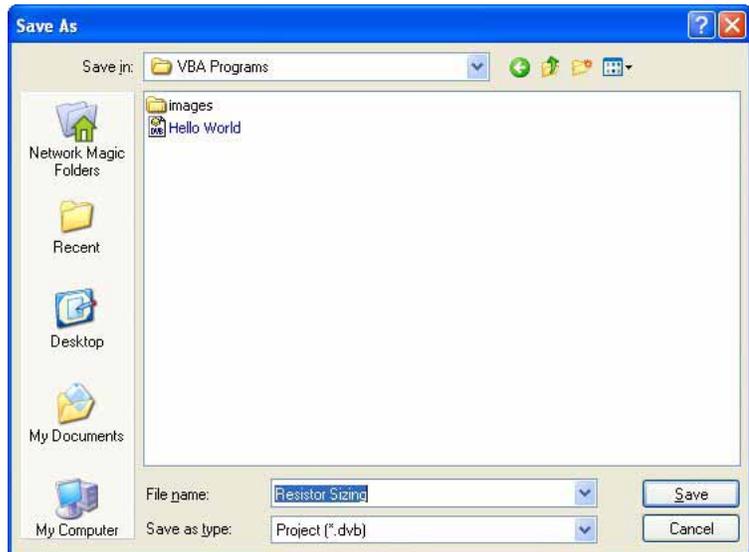


Figure A.4 – Saving the Resistor Sizing Program

Beginning a New Visual Basic Application

Remember, that all programming projects begin with one or more sketches, with one portraying the part, detail, or assembly and the other being the user input form. In this program, Resistor Sizing Program, we will just be running a user input form inside the AutoCAD application, so we only need to sketch the arrangement of this special dialogue box. In this project, we will name the input form, **Resistor Sizing Program**. We will place two textboxes on the form to type the line voltage and the current. We will place two labels for the computed resistance and power of the resistor. To the left of the textboxes and labels, we will insert the label, **Voltage, Current, Resistance** and **Power**. To the right of the textboxes and labels, we will insert the label, **volts, amperes, ohms** and **watts**. We will include an image of the two formulas. We will have three command buttons, **Compute, Reset** and **Exit**. On the bottom of the form, we will write the copyright statement using another label. On our sketch, we can help ourselves by being as accurate as possible, by displaying sizes, fonts, colors and any other specific details that will enable us to create the form quickly. From the beginning of inserting the form into the project, we need to refer to our sketch.

We should train new programmers initially in the art of form building. When using the editor, we insert and size the form, and selecting the Controls Toolbox, we will place all the various input tools and properly label them. Whenever we place an input tool, the properties window will display a list of every attribute associated with the tool, and we will make every effort to arrange the tool by performing such actions as naming, labeling and sizing the visual input device.

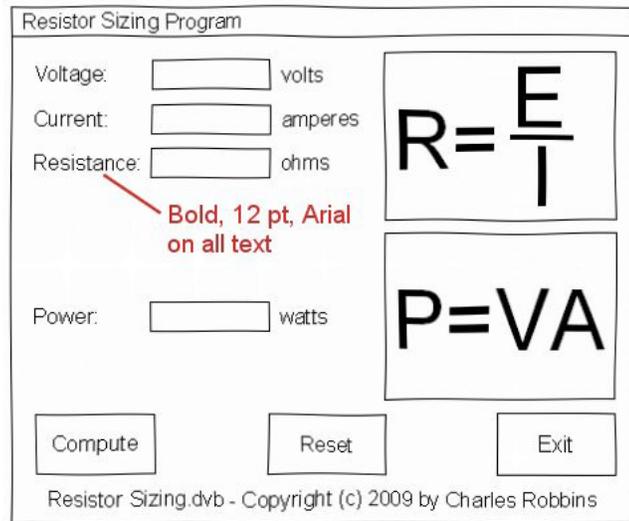


Figure A.5 – Sketch of the Resistor Sizing Form

Now that we have an idea of what the dialogue box in our program will look like, select the **Insert UserForm** button on the Standard toolbar to insert a new form. Instantaneously, the once grey work area is changed to contain the UserForm1. A Form folder with Userform1 is now in the Project menu and the Properties pane contains the attributes associated with UserForm1.

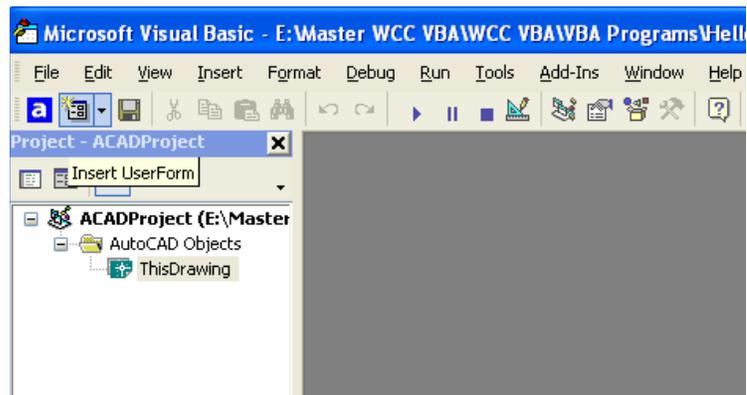


Figure A.6 – Inserting a User Form

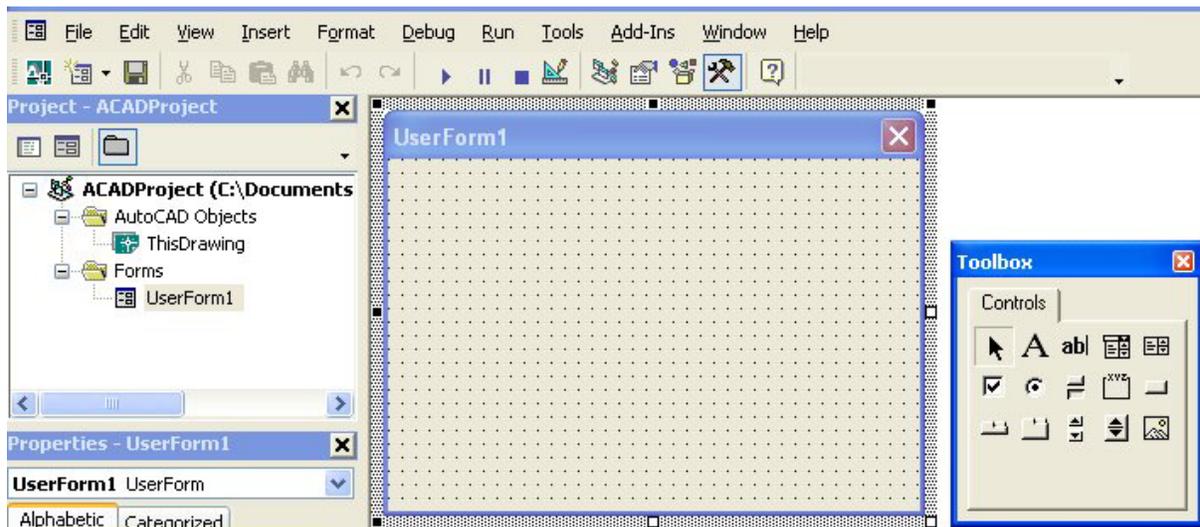


Figure A.7 – Designing the Resistor Sizing Form in Visual Basic

Laying Out a User Input Form in Visual Basic

We will change the **Caption** in the Properties pane to Resistor Sizing to agree with the sketch in Figure A.5. Go ahead and change the form in two other aspects, Height and Width.

Alphabetic	
(Name)	frmResistorSizing
BackColor	&H80000013&
Caption	Resistor Sizing Form
Height	300
Width	400

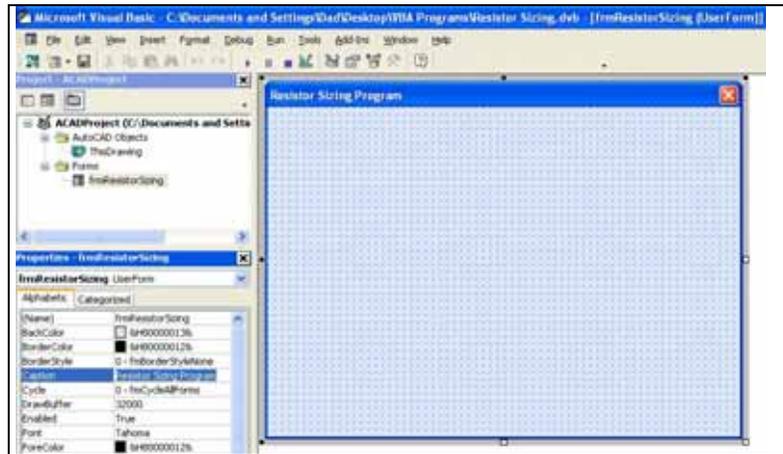


Figure A.8 – Setting the Caption and other Properties

The form will change in size to the height and width measurement. The background color will change to a light blue. There are many more attributes in the Properties pane that we will use on future projects.

Inserting a Label into a Form

A good form is easy to figure out by the user, so when we are attempting to provide information on the window that will run in AutoCAD; we add labels to textboxes to explain our intent. Press the Label (A) button on the Control Toolbar to add a label. To size the label area, click on the upper left area of the form and hold down on the left mouse button, draw the dotted label box as shown in Figure A.9.

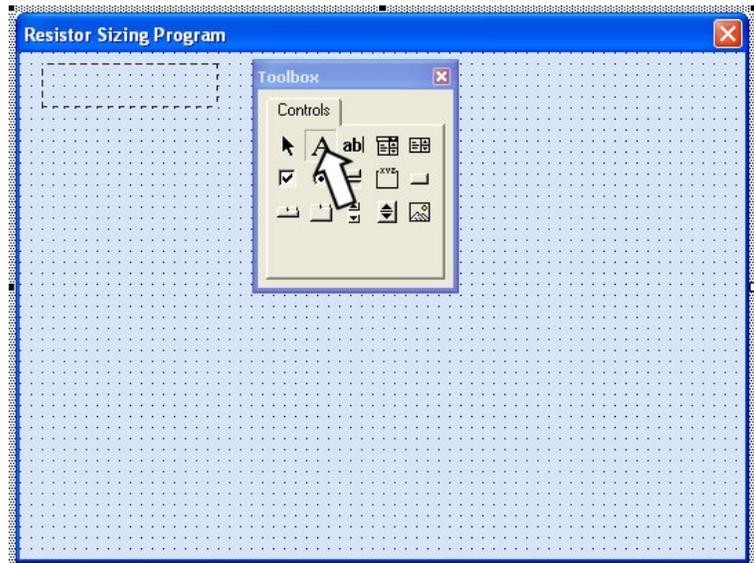


Figure A.9 – Placing a Label on the Form

We will name the Label using a common Visual Basic naming convention where the programming object is a three-letter prefix followed by the name or phrase of the tool. For our first label, the name is **lblVoltage**.

Alphabetic	
(Name)	lblVoltage
BackColor	&H80000013&
Caption	Voltage:
Font	Arial

On the sketch, the label’s caption is “**Voltage:**” The font will be a bold, 12 point, Arial. When highlighting the row for Font, a small command button with three small dots appears to the right of the default font name of Tahoma. Click on the three-dotted button to open the Visual Basic Font window.

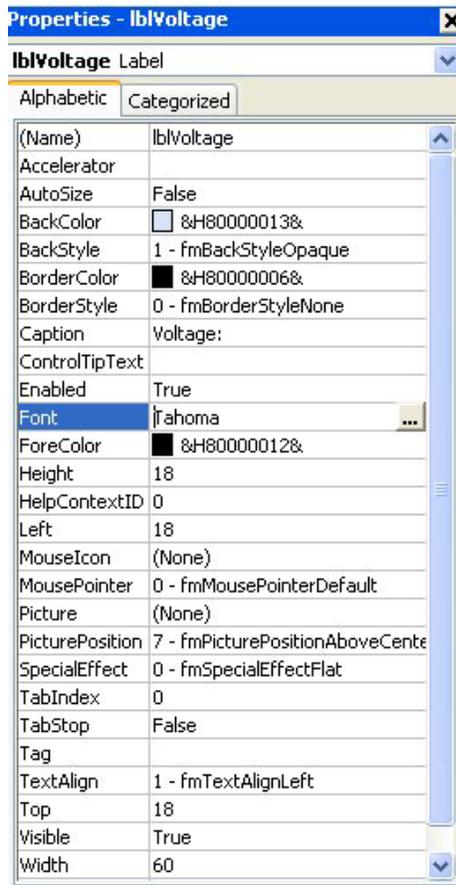


Figure A.10 – Changing the Font to Arial

We will select the Arial font, Bold font style and 12 size for this project to agree with the initial sketch if the user input form. When we adjust the attributes for the label, these changes do not alter globally for the other objects on the form. If we wish to underline the text or phrase in the label, add a check to the Underline checkbox in the Effects section of the Font window. When we finish making changes to the font property, select the OK command button to return to the work area.

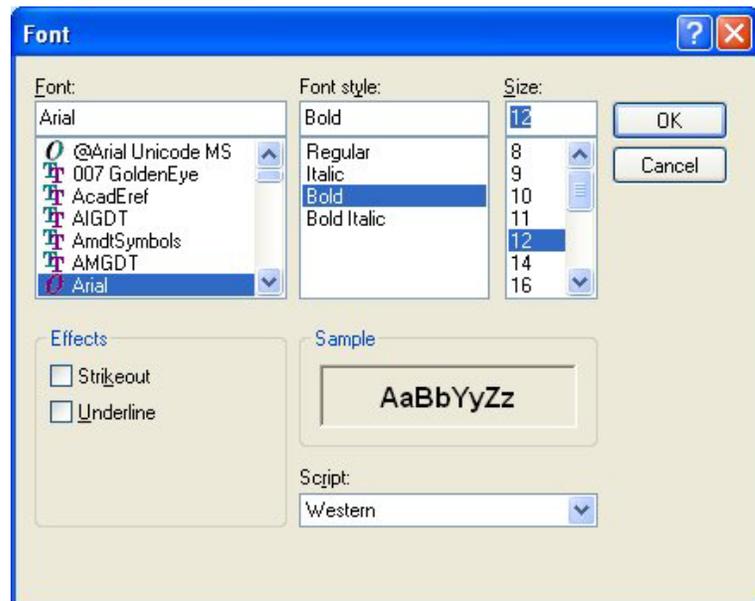


Figure A.11 – The Font Window in Visual Basic

When the first label is done, the background color of the label matches the background color of the form. In many cases, that effect is visually pleasing to the eye, versus introducing another color. Both color and shape will direct the user in completing the form along with the explanation we place on the window to guide the designer in using the automated program. Use colors and shape strategically to communicate well.

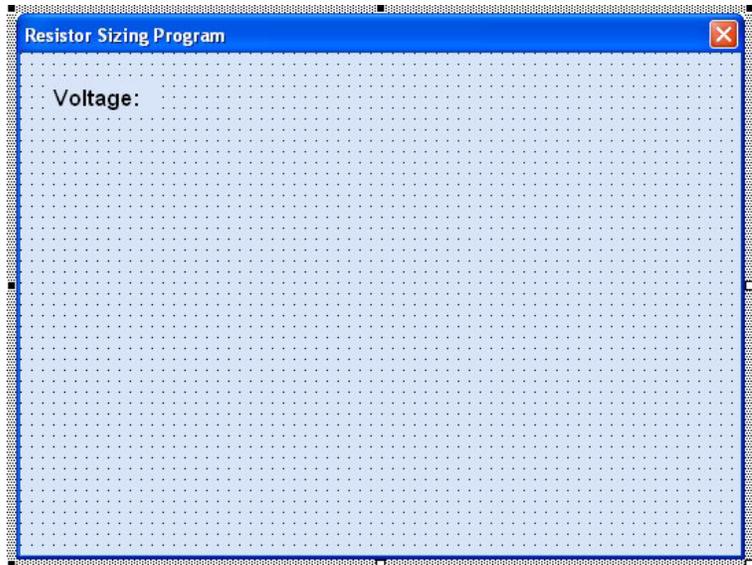


Figure A.12 – The Finished Label on the Form

Inserting a Textbox into a Form

A textbox is utilized so that a user of the computer program can input words, numbers or a mixture of both types of data strings into the form. Press the TextBox (ab) button on the Control Toolbar to add a textbox. To size the textbox area, click on the upper left area of the form and hold down on the left mouse button, draw the dotted textbox.

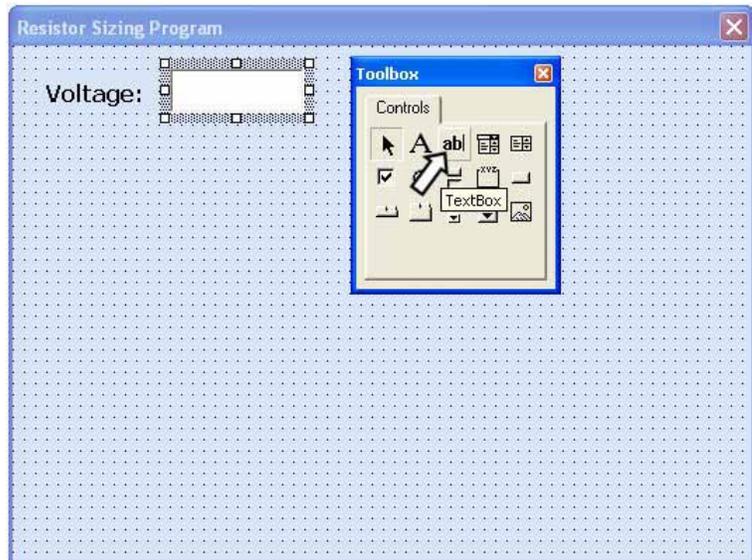


Figure A.13 – Placing a TextBox on the Form

We will name the TextBox using the three-letter prefix followed by the name or phrase of the tool. For our first textbox, the name is **txtVoltage**.

Alphabetic	
(Name)	txtVoltage
Height	24
Width	72

The font on the sketch is 12 point, Arial. When highlighting the row for Font, a small command button with three small dots appears to the right of the default font name of Tahoma. Click on the three-dotted button to open the Visual Basic Font window. Make the changes as we did on the Label and press OK to save the property.

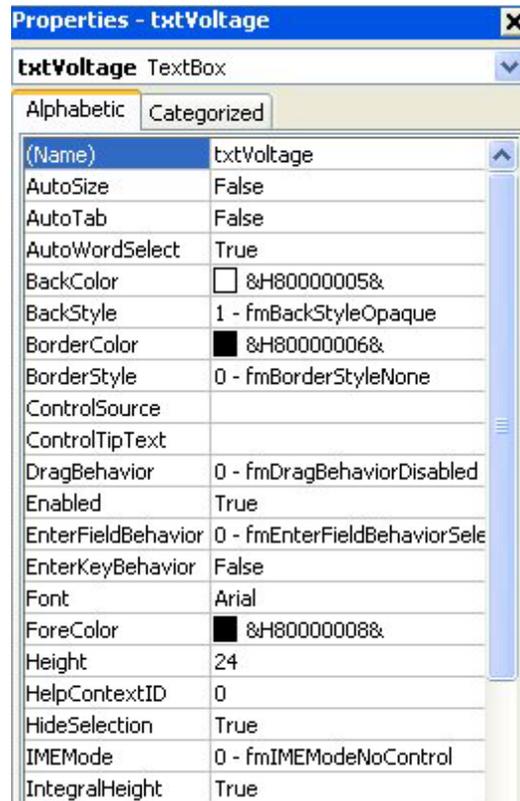


Figure A.14 – Changing the (Name) to txtVoltage

Inserting More Labels and Textboxes into a Form

We will add one more TextBox named Current under the first textbox and seven more labels as shown in the figure to the right.

The form shows the name of the labels and textboxes.

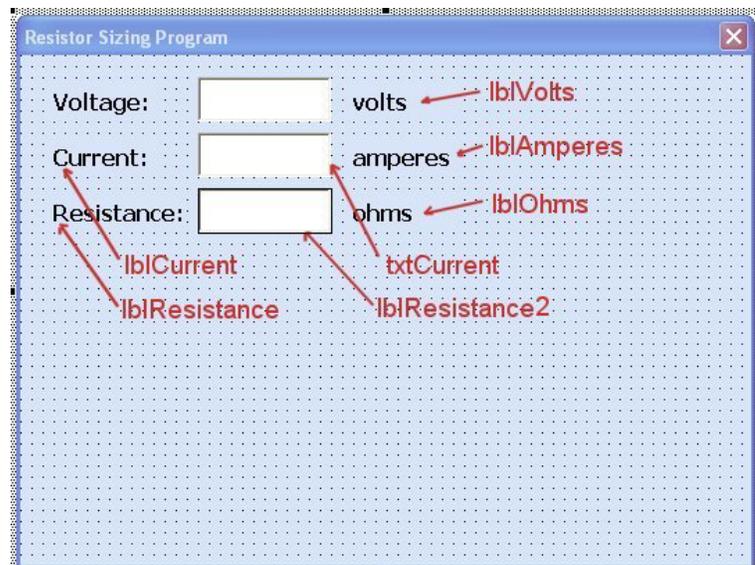


Figure A.15 – Additional Textboxes and Labels

Inserting a Label into a Form to Post the Output

Some labels on a form are in a position to display an answer after the user inputs data and they press the command button to execute the application. To add this label, press the Label (A) button on the Control Toolbar to add a label. To size the label area, click on the upper left area of the form and hold down on the left mouse button, draw the dotted label box as shown in Figure A.16.

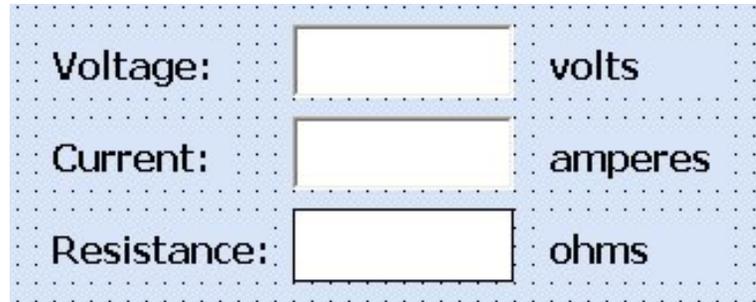


Figure A.16 – Placing another Label on the Form

We will name the Label using the name **lblResistance2**.

Alphabetic	
(Name)	lblResistance2
BorderStyle	1 - fmBorderStyleSingle
Caption	
Font	Arial
Height	24
Width	72

The font on the sketch is 12 point, Bold and Arial. When highlighting the row for Font, a small command button with three small dots appears to the right of the default font name of Tahoma. Click on the three-dotted button to open the Visual Basic Font window. Make the changes as we did before and press OK to save the property.

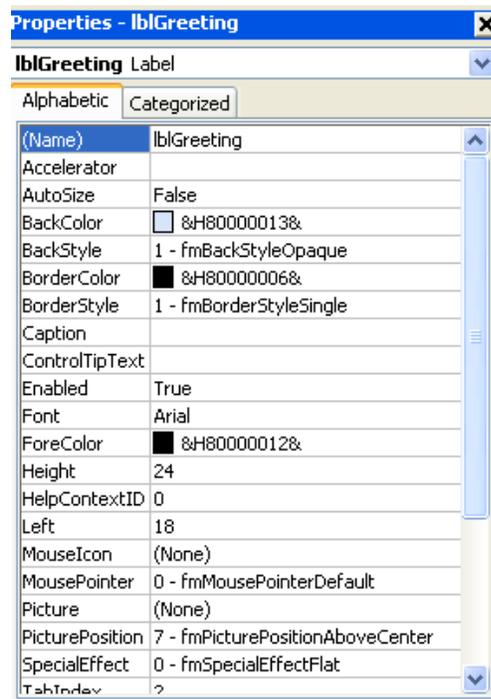


Figure A.17 – Changing the (Name) to txtName

Creating and Inserting an Image into a Form in Visual Basic

Different from the Hello World program, this form will have a picture of the formula, so we need to make a graphic in AutoCAD as shown in figure A.18. When the graphic is finished, we need to save the drawing as an image file. Use the **Saveimg** command to save file in the VBA Programs folder. Create a folder named Images in the VBA Programs folder and save the file as Formulas. We saved the file as a Bitmap with a width of 100 pixels and a height of 140 pixels.

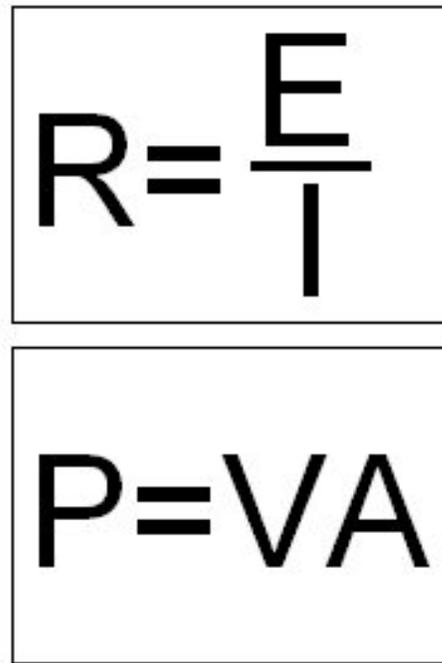


Figure A.18 – The Resistor Sizing Image File

On the control toolbox, select the Image tool and then draw a rectangular box on the form in the upper right corner as shown in Figure A.19. After outlining the size of the image, we will direct the program to the folder and filename of the digital image. In the Properties – Image pane, select the attribute named Picture. With the mouse, select the three-dot box in the empty cell to the right of Picture. The Load Picture window appears on the screen. Go to the VBA Programs folder and then the Images folder. Select the file, Stamping with four holes and it will appear in the picture frame.

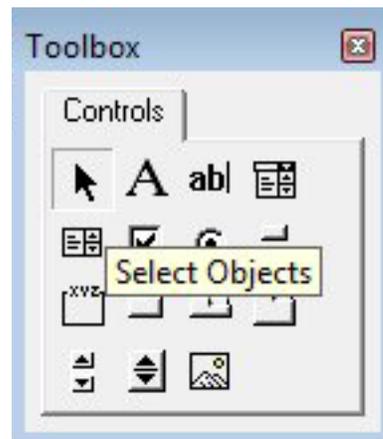


Figure A.19 – The Control Toolbox

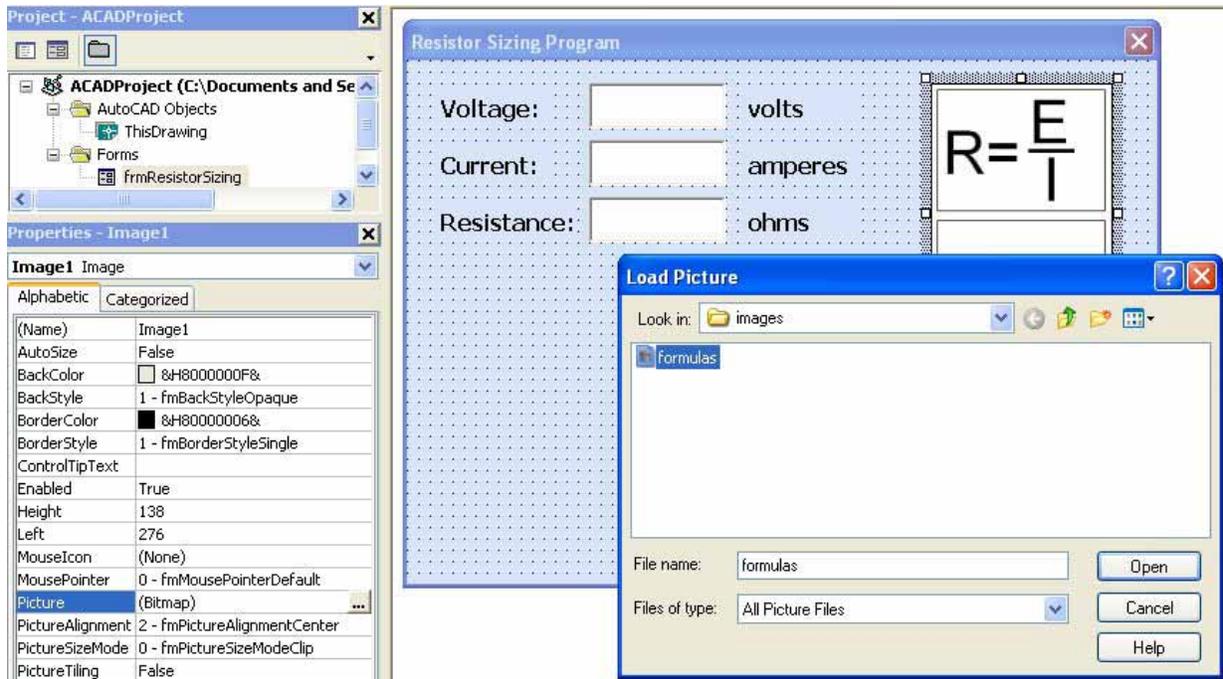


Figure A.20 – The Control Toolbox

Inserting a Command Buttons into a Form

A command button is used so that a user will execute the application. Press the Command button on the Control Toolbar to add a command button. To size the label area, click on the upper left area of the form and hold down on the left mouse button, draw the command button as shown in Figure A.21.

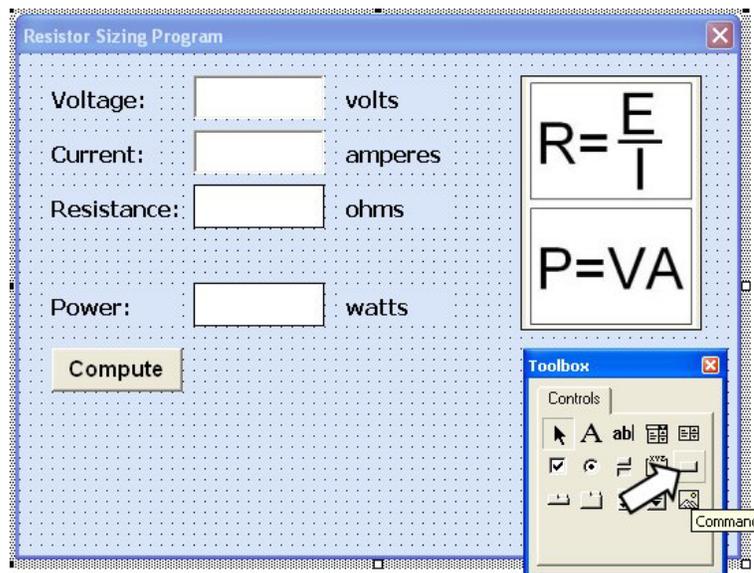


Figure A.21 – Insert a Command Button onto a Form

We will name the command button using the name is **cmdCompute**.

Alphabetic	
(Name)	cmdCompute
Caption	Compute
Font	Arial
Height	24
Width	66

The font on the sketch is 12 point, bold and Arial. When highlighting the row for Font, a small command button with three small dots appears to the right of the default font name of Tahoma. Click on the three-dotted button to open the Visual Basic Font window. Also, we will center the text on the command button. Make the changes as we did before and press OK to save the property.

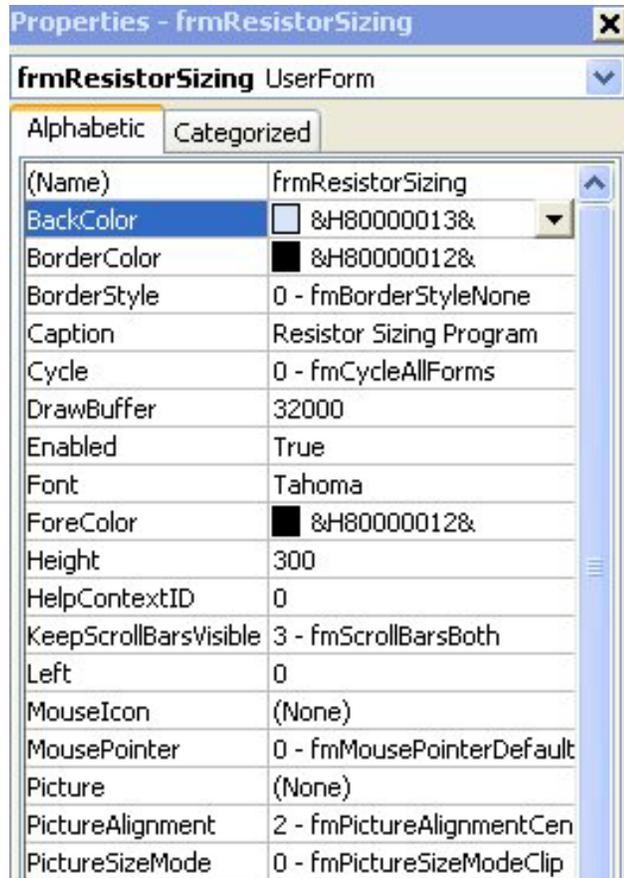


Figure A.22 – Changing the (Name) to cmdCompute

Add a second Command button, named cmdReset is for clearing the txtVoltage, txtCurrnet, lblResistance2 and lblPower2 objects. The third command button is to exit the program. When the user presses the Exit command button, the application closes and full control of the manual AutoCAD program returns to the user. Notice the equal spacing between the command buttons gives a visually friendly appearance.

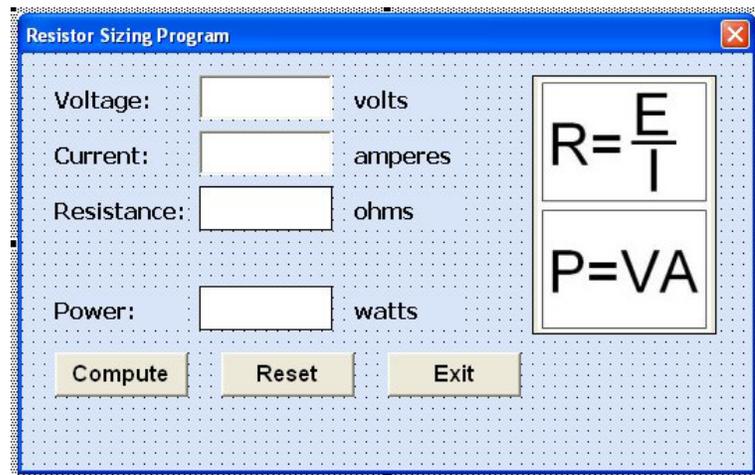


Figure A.23 – Insert Two More Command Buttons

Adding a Copyright Statement to a Form

At the beginning of a new program, we will expect to see an explanation or any special instructions in the form of comments such as copyright, permissions or other legal notices to inform programmers what are the rules dealing with running the code. Comments at the opening of the code could help an individual determine whether the program is right for their application or is legal to use. The message box is a great tool when properly utilized to inform someone if they are breaking a copyright law when running the code.

Finish the form with the following copyright information.

```
'Resistor Sizing.dvb copyright (c)  
2009 by charles robbins
```

If there are special rules or instructions that the user needs to know, place that information on the bottom of the form.

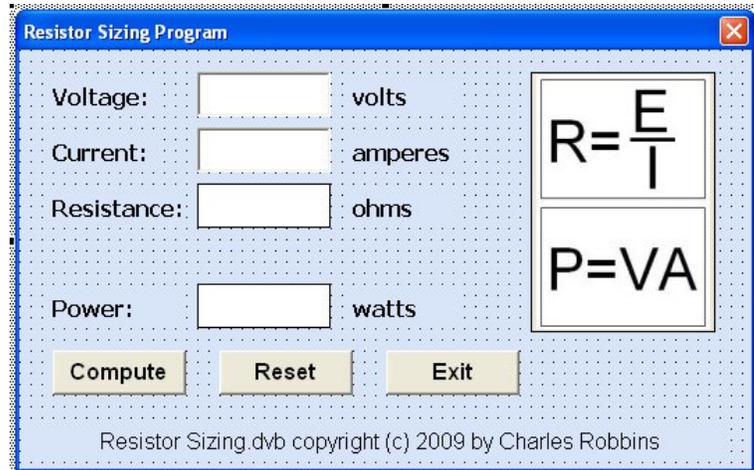


Figure A.24 – Adding a Copyright Statement

Adding Comments in Visual Basic to Communicate the Copyright

The comments we placed in the first three lines of the program will inform the individual opening and reading the code of any special instructions. However, for those user that may run the application without viewing the internal comments, the label on the bottom of the form with the copyright information is a great tool to alert the client to the rules of the program

To begin the actual coding of the program, double click on the Compute command button. At the top of the program and before the line of code with Private Sub cmdCompute_Click (), place the following comments with the single quote (') character. Remember, the single quote character (') will precede a comment and when the code is compiled, comments are ignored.

Type the following line of code:

```
'Resistor Sizing.dvb copyright (c) 2009 by Charles W. Robbins  
'This program will open a dialogue box in AutoCAD, allow the user to type the line voltage and  
'current. When the user clicks on the Compute button, the Resistance and Wattage is given
```

```

cmdCompute Click
'Resistor Sizing.dvb copyright (c) 2009 by Charles W. Robbins
'This program will open a dialogue box in AutoCAD, allow the user to type the line voltage and
'current. When the user clicks on the Compute button, the Resistance and Wattage is given

Private Sub cmdCompute_Click()

End Sub

```

Figure A.25 – Adding a Copyright Statement

Programming the Compute Command Button

We can type these two Visual Basic expressions to compute the resistance and power for the resistor.

```

'Resistor Sizing.dvb copyright (c) 2009 by Charles W. Robbins
'This program will open a dialogue box in AutoCAD, allow the user
'current. When the user clicks on the Compute button, the Resistor

Private Sub cmdCompute_Click()
lblResistance2 = txtVoltage.Text / txtCurrent.Text
lblPower2 = txtVoltage.Text * txtCurrent.Text
End Sub

```

Figure A.26 – Declaring Variables with Dim Statements

lblResistance2 = txtVoltage.Text / txtCurrent.Text
lblPower2 = txtVoltage.Text * txtCurrent.Text

Resetting the Data

To clear the textboxes for the voltage and current and to reset the labels containing the resistance and power, we will set the textbox for Voltage, txtVoltage.text property to a blank entry by using the equal sign “=” and the null string “”. This makes the property blank. We will set the label for Resistance, lblResistance.caption property to a blank entry by using the equal sign “=” and the null string “”, and this will make that property blank. Do the same for txtCurrent and lblPower2.

```

'Resistor Sizing.dvb copyright (c) 2009 by Charles W. Robbins
'This program will open a dialogue box in AutoCAD, allow the user
'current. When the user clicks on the Compute button, the Resistor

Private Sub cmdCompute_Click()
lblResistance2 = txtVoltage.Text / txtCurrent.Text
lblPower2 = txtVoltage.Text * txtCurrent.Text
End Sub

Private Sub cmdReset_Click()
txtVoltage = ""
txtCurrent = ""
lblResistance2 = ""
lblPower2 = ""
End Sub

```

Figure A.27 – Computing the Reset Button

Exiting the Program

```
'Resistor Sizing.dvb copyright (c) 2009 by Charles W. Robbins
'This program will open a dialogue box in AutoCAD, allow the user to type the line voltage
'current. When the user clicks on the Compute button, the Resistance and Wattage is given

Private Sub cmdCompute_Click()
lblResistance2 = txtVoltage.Text / txtCurrent.Text
lblPower2 = txtVoltage.Text * txtCurrent.Text
End Sub

Private Sub cmdExit_Click()
'Unload and exit the program
Unload Me
End
End Sub

Private Sub cmdReset_Click()
txtVoltage = ""
txtCurrent = ""
lblResistance2 = ""
lblPower2 = ""
End Sub
```

Figure A.28 – Exiting the Program

To exit this program, we will unload the application and end the program. Type the following code:

```
"Unload and exit the program
  Unload Me
  End
```

Running the Program

After noting that the program is saved, press the F5 to run the Resistor Sizing application. The Resistor Sizing Program window will appear on the graphical display in AutoCAD. Notice the professional appearance and presentation of information in a clean dialogue box. The color of the background is neither black nor white, which would match the normal graphical display colors used by designers in AutoCAD.

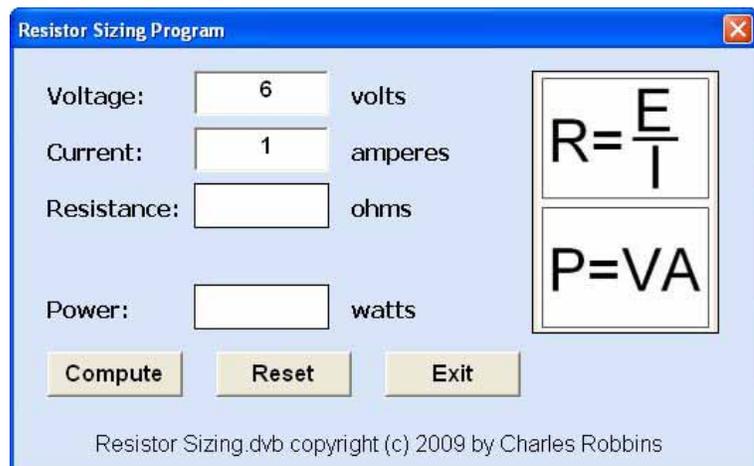


Figure A.29 – Launching the Program

We can type 6 in the voltage textbox and 1 in the current textbox. If we press the Reset command button, the textboxes and labels will clear for new values. If we push the Exit command button, the program will close. In this case, we will press the Compute command button and the answers appear as shown in figure A.30.

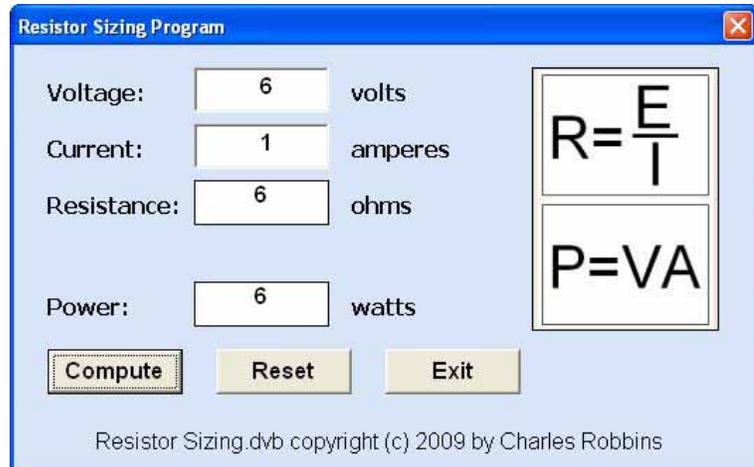


Figure A.30 – Running the Program

If our program does not function correctly, go back to the code and check the syntax against the program shown in Figure A.28. Repeat any processes to check or Beta test the program. When the program is working perfectly, save and close the project.

We can make many variations of this Visual Basic Application using other formulas. While we are practicing with forms, we can learn how to use textboxes, labels and comments. These are skills that we want to commit to memory.

*** World Class CAD Challenge 5-11 * - Write a Visual Basic Application that displays a single input form, allow the user to type in their name, and when executed, the program will greet the user with information obtained from the computer.**

Continue this drill four times using some other form designs, each time completing the Visual Basic Application in less than 30 minutes to maintain your World Class ranking.