TOE

Project 1 was a very simple application. In planning projects as simple as this we can use a method called TOE – tasks, objects, events:

tasks – identify what we want to do

objects – choose the components we will use to perform these tasks

events – decide which events will trigger the objects to perform the tasks.

(It is sometimes also a good idea to prepare a sketch of where we will place the components on the screen when we do the O part of TOE.)

In Project 1 the task was to turn the form red, the object used was a command button, and the event was a mouse click on the button.

The TOE method will be used for simple projects. When we get onto more complex programming we will need more sophisticated methods, but even then the TOE method can be used for each form.

Project 2 – Displaying a message

In this next exercise we will see how to display a message on screen and, by so doing, find out more about the basic elements of VB.

We are going to create the following screen that will display and hide a message:

Task: Hide or display a simple message.

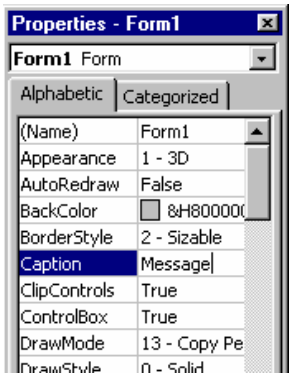
Objects: Label, textbox and three command buttons.

Events: Click on show button to display message, click on hide button to hide message, click on exit button to end application.

1. Start VB and create a new standard EXE project.(If you are already in VB with a project open, from the top menu choose *File / Remove Project* first or your new project will be added to the last!)

   Make sure that at least the form, the properties window and the toolbox of the IDE are visible.

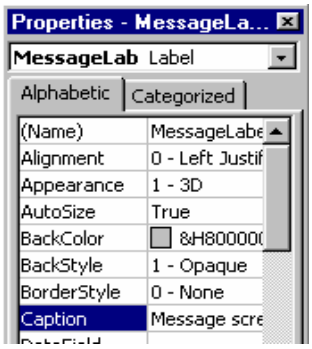
2. Click once anywhere on the form to select it, and then in the properties window write *Message* next to *Caption*. The caption *Message* will appear at the top of the form (see at right).



3. a On the toolbar double click the label button  . Drag the label that appears to the top left corner of the form.

   b Make sure the label is selected (sizing handles are visible) and go to the properties window and set the following properties:

Name:	MessageLabel	(no spaces)
AutoSize:	True	(double click to change)
Caption:	Message Screen	(space between words)
Font:	(*choose your own*)	(click on the ellipsis **...** to alter font details)



The *AutoSize* will get the label to automatically resize to fit the caption.

4. We will now add three command buttons for *Hide, Show* and *Exit*.

   a Position the three buttons by double clicking them onto the form and dragging them to the correct location (see sample screen on previous page).

   b Set the properties of the buttons as follows:

	Command1	*Command2*	*Command3*
Caption:	&Show	&Hide	E&xit
Name:	ShowButton	HideButton	ExitButton

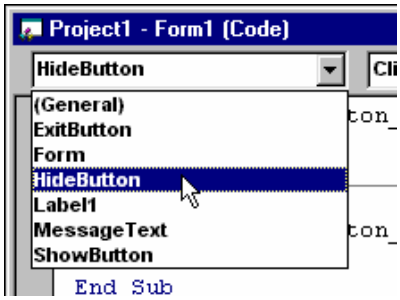
The *&* shows up as an underline and will allow a keyboard command (e.g. *Alt* + *X* for exit).

5. a Now add the *TextBox*  for the message to appear in, and drag it into position.

   b Set the following properties for the box:

Name:	MessageText
AutoSize:	True
Font:	(*choose one*)
Text:	(*delete*)

6. We are now ready to write the Basic code to go with this form.



To do this double click on each button to jump to the code editor at the correct place in the program.

After writing each line either return to the form and double click on the next button, or use the drop down list at the top left of the code editor.

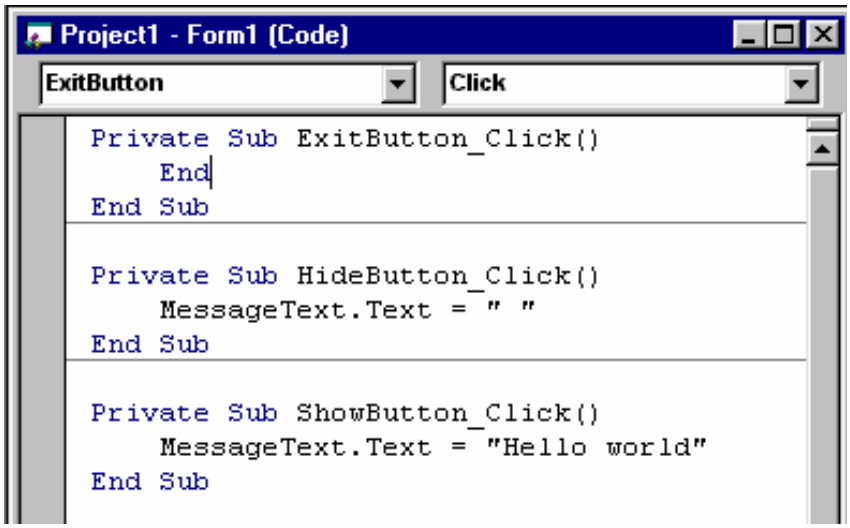
Write the following for each button in turn in the empty line between `Private Sub` and `End Sub`:

ShowButton: `MessageText.Text = "Hello world"`

HideButton: `MessageText.Text = " "`

ExitButton: `End`

When finished your code editor should look like the following:



Notes:

- there are no spaces in `MessageText.Text`
- the text for *HideButton* is two double quotes " " (*this will display a blank*)
- = is used to place a value in the *Text* property of the *MessageText* control when the program is running
- at times VB attempts to guess what you are going to type and attempts to auto-complete it for you (see at right); if it does this the *Tab* key will accept the selection and move on; *Enter* will accept the selection and move to the next line

- if you start each line with a *Tab* it will indent the code correctly, making it more readable

- remember the golden rule in using the code editor is to never alter anything VB has written for you.

7. When ready save the project  in the *Project 2* folder you created earlier. Call the form *Message* and the project *Project2*.

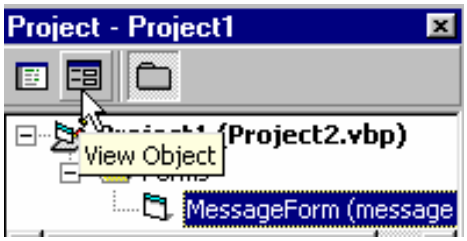
8. Run the program with the  button or by pressing the F5 key.

   When you click on *Show* the message will appear. *Hide* will overwrite the message with nothing (“ “); and *Exit* will quit. Note also the Alt+S, Alt+H, and Alt+X keys will also work.

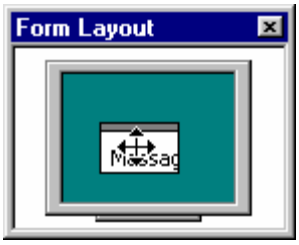
   If you have made a mistake the code editor window will appear with an error message, indicating the section of code where the error occurs. If necessary press the *Help* button for more information on the mistake.

9. Alter your program in the code editor so that a different (but appropriate!) message appears.

   Remember an easy way to move between the form and the code editor is to use the *View Code* and *View Object* buttons in the project explorer.



   You can also change where the form is placed using the *Form Layout* window. (If this is not visible click on the form layout button on the standard toolbar.) In this window you can click on the form and drag it to the part of the screen you wish it to be displayed.



   Alternatively you might also like to explore the *WindowState* property of the form so that it appears maximised when run.

10. a On the form click anywhere that is not a button or label to select the form, and in the properties window experiment with *BackColor*.

    b Repeat for the label at the top.

    c You may also like to alter the colour and font of the text on the buttons.

    d Reposition the objects on the screen to suit yourself.

11. When satisfied re-save the project and exit.