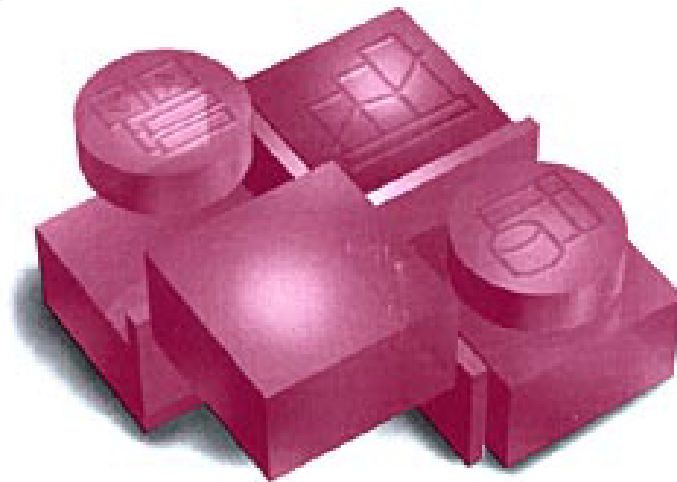


Standard Grade Computing Studies

Pupil Notes Book 2



Microsoft
Visual Basic 6.0
Professional Edition

Visual Basic

Do

Counters

Here's a quick note on counting things up in Visual Basic.

You might want to keep track of how many questions that have been answered correctly in a quiz, or keep a total of numbers which have been entered. For both of these you need a counter.

The following code adds 1 to a counter each time the cmdAdd button is clicked:

```
Option Explicit          ' General Area
Dim counter As Integer  ' Any name will do, it doesn't need to be counter
.....
Private Sub cmdAdd_Click()
counter = counter + 1    ' Adds 1 to the value of the variable 'counter'
lblDisplay.Caption = counter    ' Displays value of counter on a label
End Sub
```

Notes:

The variable 'counter' is set up in the general area, because if it was in the cmdAdd procedure, it would be reset to zero every time the button was clicked.

Read the line - "counter = counter + 1"

As - "The new value for counter = the old value for counter + 1"

Other counters will look like this:

Total = Total + money

Or

Number = Number + VAT

In all counters there is a pattern. **xxx = xxx + yyy**

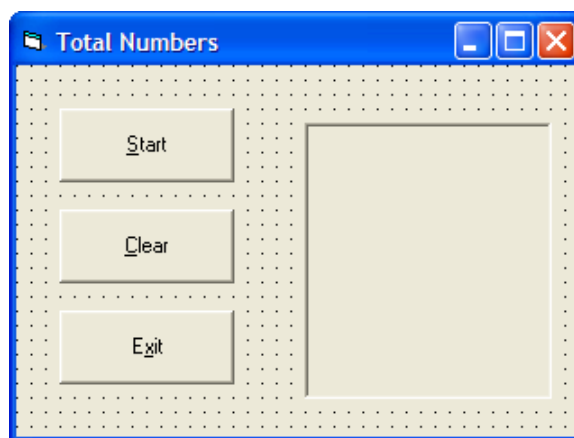
The two items in place of **xxx** must be the same. The **yyy** item is what you are adding on each time. It can be a number, like 1, or a variable, like 'number'.

Do

Running Total

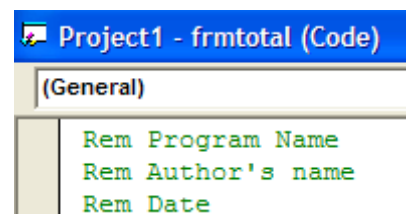
For the program you are going to produce, *you can use the code shown below as a guide*. The program that you are about to produce will calculate the total value of several numbers entered by the user. *However, the number of numbers the user enters may vary from one program run to the next.*

1. Open the form **frmTotal** from your Visual Basic folder.



The properties for all the objects and the code for the Exit and Clear commands have been added. All we have to do is alter the program, author and date details and add the code to the Start button.

2. Change the three REM lines to include a suitable program name. Add your details as the author and put today's date in the date line.



3. Add the following code to the **Start** button between Sub and End Sub.

```
Private Sub cmdStart_Click()  
Dim Times, Counter As Integer, Number, total As Single ' Set  
variables  
Times = InputBox("Enter the number of times you wish to enter a  
number", "number of loops") ' how many numbers you wish to  
add together  
  
For Counter = 1 To Times 'begin loop  
Number = InputBox("please enter a number", "numbers") ' get user  
numbers  
total = total + Number ' add user number to running total  
Next  
picDisplay.Print "your total is "; total ' display total  
End Sub
```

4. Save and Run your program making sure all the buttons work.

Do

Rocket Launch

Develop a program that will act as a countdown timer for Space Shuttle launches at Cape Canaveral. The program will make use of a FOR ... NEXT loop and a counter. The counter will decrement by one instead of incrementing by one. The program will then display the message "We have blast off" in a messagebox.

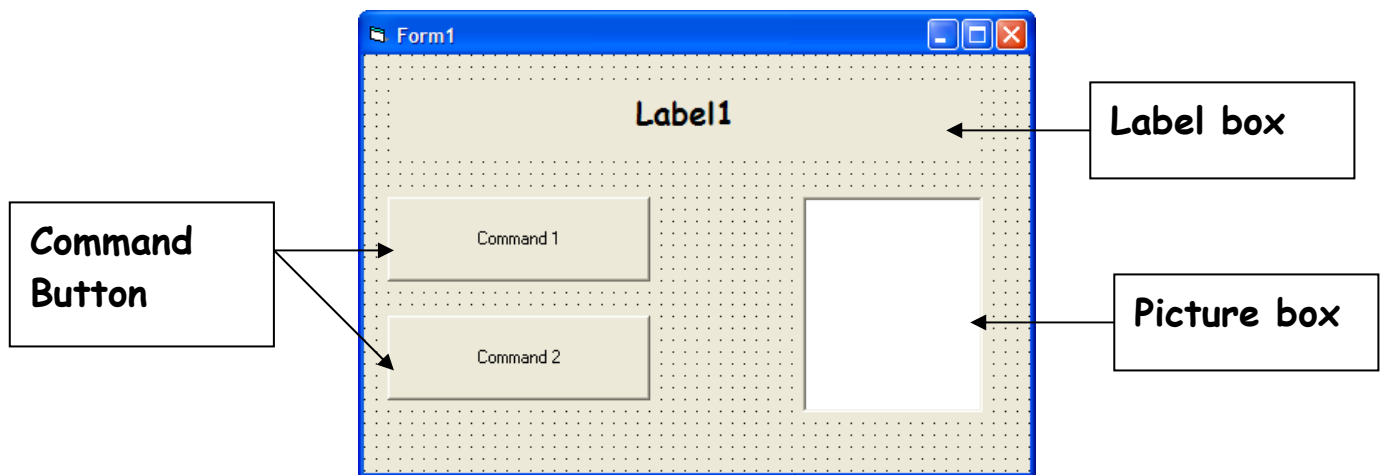
```
For Countdown = 10 to 1 step - 1  
MsgBox "Blast off in " & Countdown & " seconds"  
Next Countdown  
MsgBox "We have blast off"
```

Do

Disappearing Clown

So far most of the programs you have created have been quite simple and have not been very eye-catching. We are going to create a program which includes a graphic.

1. Create a simple form as shown below.



2. Set the properties of each object as shown below.

Object	Property	Value
Form	Name	frmClown
	Caption	Disappearing Clown
Label	Name	lblInfo
	Caption	Blank
	Alignment	2 - Center
Command 1	Name	cmdShow
	Caption	Now you see me
Command 2	Name	cmdHide
	Caption	Now you don't
Picture box	Name	picDisplay
	Visible	False

We are going to insert a graphic in the Picture box.

3. Select the **PicDisplay** object. (handles)
4. In the properties box scroll down to Picture.

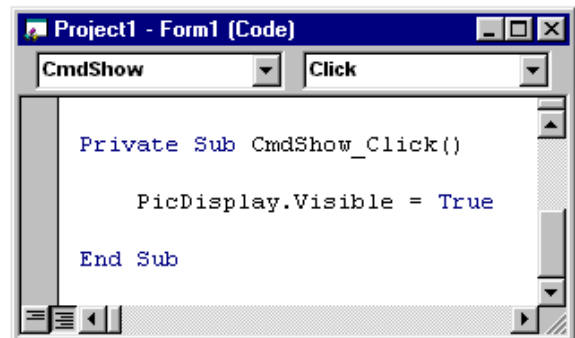
5. Select the button to the right of none.



6. Go to your Visual Basic folder and select the clown file. This should add a clown picture to the picture box.

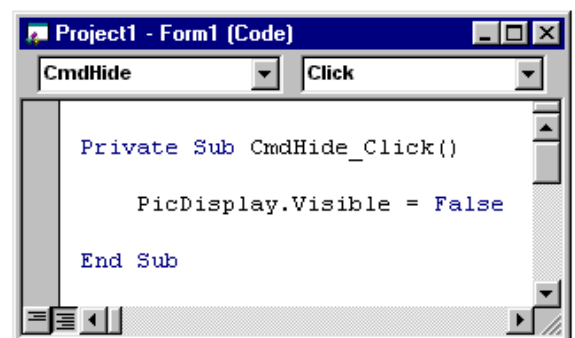


7. Add the following code to the **cmdShow** button.



```
Project1 - Form1 (Code)
CmdShow Click
Private Sub CmdShow_Click()
    PicDisplay.Visible = True
End Sub
```

8. Add the following code to the **cmdHide** button.



```
Project1 - Form1 (Code)
CmdHide Click
Private Sub CmdHide_Click()
    PicDisplay.Visible = False
End Sub
```

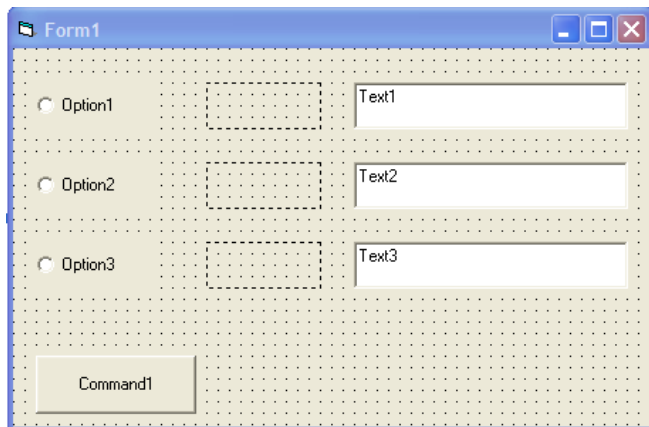
9. Save the form and project as *frmclown* and *prjctclown*.
10. Run your program to see if it works.

Do

Face Options

The program you are about to design will be used by teachers to design stickers for pupil effort in completing homework tasks. It will incorporate three graphics and suitable quotes. The program will allow the teacher to choose any one of three option buttons for printing

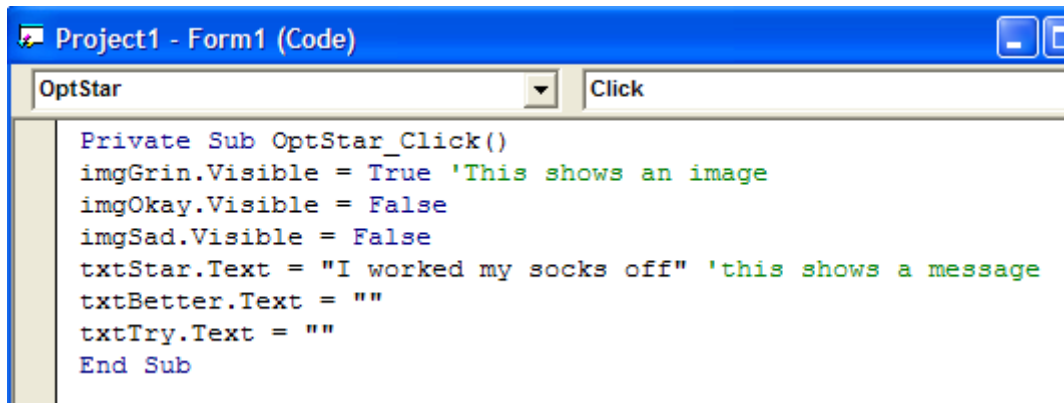
1. Open the form **frmFaces** from your Visual Basic folder. You should see the following form.



2. Change the properties of all the objects as listed in the Property grid shown below.

Object	Property	Setting
Form	Caption	Sticker Program
	Name	frmStickers
Option1	Caption	You're a Star Pupil!
	Name	OptStar
Option2	Caption	Keep up the good work!
	Name	optBetter
Option3	Caption	Try harder next time!
	Name	optTry
Image 1	Name	imgGrin
	Picture	Face1
	Visible	False
Image 2	Name	imgOkay
	Picture	Face2
	Visible	False
Image 3	Name	imgSad
	Picture	Face3
	Visible	False
Text 1	Border Style	None
	Name	txtStar
	Text	None
Text 2	Border Style	None
	Name	txtBetter
	Text	None
Text 3	Border Style	None
	Name	txtTry
	Text	None
Command 1	Caption	E&xit
	Name	cmdExit

3. Click on the option button **optStar** and add the code as shown below.



```
Private Sub OptStar_Click()  
imgGrin.Visible = True 'This shows an image  
imgOkay.Visible = False  
imgSad.Visible = False  
txtStar.Text = "I worked my socks off" 'this shows a message  
txtBetter.Text = ""  
txtTry.Text = ""  
End Sub
```

4. Click the other option buttons and add the code. Use the above code and work out the changes required. Think of two suitable messages for each button.
5. Save your project and form with suitable names.
6. Demonstrate your working program to your teacher.

Do

P7 Subject Option

Each year Primary 7 pupils visit the school and participate in some subjects. You have been asked by the head teacher to develop a program that will instruct the primary guides as to which subject each primary school should attend. The selection of each primary will result in directions being displayed and a suitable graphic for that subject.

Design a program for the head teacher to evaluate.

Read

IF Statements

When you have to make a decision or choice in programming you use an **IF Statement**.

Imagine a pupil had done an exam and the teacher wanted to display a message depending on the pupil's mark. Just to confuse you, the 3 different types of IF statements are written as follows:

(Simple)

```
IF total > 49 THEN picDisplay.Print "You have passed"
```

(Normal)

```
IF total > 49 THEN  
    PicDisplay.Print "You have passed"  
ELSE  
    PicDisplay.Print "You have failed"  
END IF
```

(Complex)

```
IF total > 69 THEN  
    PicDisplay.Print "You have passed at Grade A"  
ELSEIF total > 59 THEN  
    PicDisplay.Print "You have passed at Grade B"  
ELSEIF total > 49 THEN  
    PicDisplay.Print "You have passed at Grade C"  
ELSE  
    PicDisplay.Print "You have failed"  
END IF
```

Although 3 ways are shown here, you will normally use the middle one. No matter which kind you use, you must follow the pattern used above, otherwise you will get a **SYNTAX error**, because you will have broken the Visual Basic rules.

You must have the IF, THEN, ELSE, (ELSEIF) and END IF commands in the right place!

Golf - Open Qualification

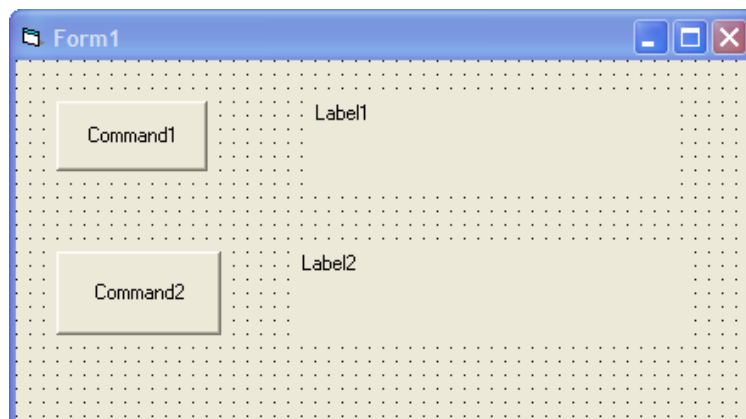
We are going to design a program that will allow a golfer to enter their four qualification scores. The program will then calculate their average score and display this average and a message to tell them if they have qualified for the Open Championship. The program will have two command buttons, End & Calculate and two labels to display the average score and a suitable message on qualification.

Program Design

Problem (In my own words)

I have to design a program that will ask the golfer to enter 4 scores. The program will add each score together and then calculate the average score for the four rounds. The program will display the average score and display a message informing the golfer if they have qualified for the Open Championship.

Interface Design



Property Grid

Object	Property	Value
Form	Name	frmOpengolf
	Caption	Open Golf Championship
Command Button	Name	cmdScore
	Caption	&Score
Command Button	Name	cmdExit
	Caption	E&xit
Label	Name	lblAverage
	Caption	(blank)
Label	Name	lblQualify
	Caption	(Blank)

Algorithms for Each Button (Procedure)

Start

1. Set up variables
2. Repeat 4 times
3. Prompt user for golf score
4. Calculate the running total
5. Calculate average score
6. Display average score and message

End

1. End program

Refinements

- 3.1 Display input box prompting user to enter their score
- 3.2 Assign user score to variable score

4.1 value of variable total equals variable total + variable score

5.1 Divide variable total by 4 and assign to variable average

6.1 Display value of variable average in label score and message in label message

Evaluation

I found this program easy as I have completed programs that were similar with loops and if statements previously. All I had to do was to take these skills and adapt them to suit the program requirements. If I had to improve my program I would create a nicer user interface with graphics or coloured text and prompt the user if they didn't enter a number.

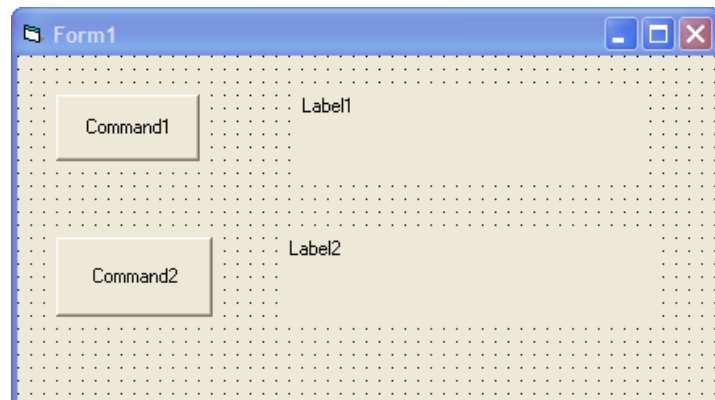
You now, are going to create the program above from design through to evaluation using the sheets provided.

1. Copy the problem statement, design grid and properties table shown above.
2. Create the HCI, change the object properties and add the code.

Show your teacher your design sheets.

Do

3. Create your form with two command buttons and two labels. As shown.



4. Change the properties of each object including the form as per your design.
5. Double-click on the **cmdExit** command and add the code to end the program.

6. Double-click on the **cmdScore** button and add the following code.

```
Private Sub cmdScore_Click()  
Dim score As Integer  
Dim total As Integer  
Dim average As Integer  
For loops = 1 To 4  
score = InputBox("enter number")  
total = total + score  
Next  
average = total / 4  
If average < 70 Then  
lblQualify.Caption = "You qualified, Well done"  
Else  
lblQualify.Caption = "Sorry you didn't qualify"  
End If  
lblAverage.Caption = "Your average score is:  " & average  
  
End Sub
```

Note: Change the label names to suit the properties of **your** labels.

7. Save your project and form with a suitable name.
8. Run your program to see if it works.
9. Demonstrate your working program to your teacher.

Copy the School Heating assessment from the PrepWork folder into your own folder.

PRACTICAL ABILITIES TASK

SCHOOL HEATING

Your teacher will outline how you should do this task and give you opportunities for discussion. Read and follow all instructions carefully, use hardware and software properly and write your report neatly.

The heating system in a school should be switched on if the average temperature is less than 17 degrees Celsius ($^{\circ}\text{C}$). The average temperature is found from the temperatures in the Art, English and Music Departments. You are required to write a program that allows the user to input the three temperatures. The program calculates and displays the average temperature then displays 'Heating should be on.' or 'Heating should be off.' as appropriate.

1 DESCRIBE METHOD - Analysis

3 marks

Show that you understand what is required by describing how you will do this task.

2 LIST STEPS - Design

6 marks

On a separate sheet of paper, show the main steps and refinements.

3 ENTER PROGRAM - Implementation

5 marks

Enter the program listing. Include internal commentary. Correct any mistakes that you make. Save the listing.

4 TEST PROGRAM - Testing *5 marks*

Complete the table below, by entering test data. You should enter sets of three temperatures then calculate the average and

indicate whether heating should be on or off. The first set should have an average below 17°C, the second set an average of exactly 17°C and the third average should be above 17°C.

Complete the last two columns by recording the program's output with your chosen test data.

	Art temperature	English temperature	Music temperature	calculated average temperature	calculated on or off	program's average temperature	program's on or off
1							
2							
3							

Compare your calculations with the program's output and write an appropriate comment.

5 GET PRINTOUTS - Implementation

2 marks

Get **two** printouts with your name in the footer. One should show your listing and the other a run using one of your sets of test data.

6 SUGGEST IMPROVEMENT - Evaluation

2 marks

Write down **one** way in which your program could be better.

7 JUDGE PERFORMANCE - Evaluation

2 marks

Read this task sheet again. Describe how well you think you have done the task.

Grade 3 - 18 or more

Grade 4 - 17 to 13

Grade 5 - 12

Grade 7 - 11 or less

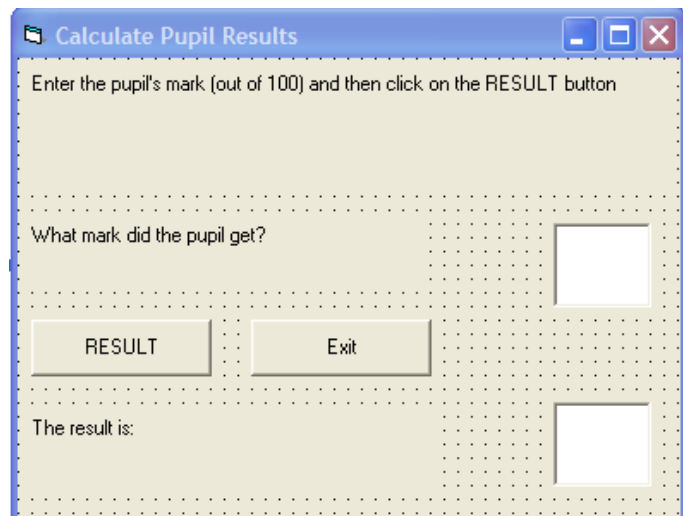
Do

Pass or fail?

The following form and program code relate to a program that first asks the user to enter a student's test mark, then determines whether the student has passed or failed the test, finally displaying the verdict.

1. Create a new form with 2 command buttons, three labels and two text boxes.

2. Set the properties of each object including the form.



3. Add the following code to **cmdResult** button.

4. Save the form and project with suitable names.

5. Test your program to see if it works.

6. Show you working solution to your teacher.

```
Private Sub CmdCalcResult_Click()

REM Short program description
REM Author's Name
REM Date
Dim Mark As Integer, Result As String
Mark = Val(TxtMarkIn.Text)
If Mark >= 50 Then
Result = "Pass"
Else
Result = "Fail"
End If
TxtResultOut.Text = Result

End Sub
```

Do

Password Program

Design a program that asks the user for a password. If the password is correct the program will display a suitable welcoming message and if wrong, will print a warning.

Hint: You should use the Results program as a basis for your new program.

Read

Random Numbers

Visual Basic can produce random numbers in the same way that you can. If a magician asks you to think of a number between 1 and 100, then you would pick a number *at random*.

To get Visual Basic to pick a random number we have to use a set method.

First you must use the word `RANDOMIZE` on its own, this tells the computer to reset its random number generator.

Next you use a formula along with a variable. e.g:

```
Private Sub CmdFindNumber_Click ()
```

```
Dim number as integer    ' use a variable called number to  
store it
```

```
Randomize                ' make the number more random
```

```
number = Int(10 * Rnd + 1) ' pick a number, store it in the  
variable
```

```
lblRandom.caption = number ' show the number on a label
```

```
End Sub
```

This example will choose a number at random between 1 and 10 every time a button called `CmdFindNumber` is clicked.

If you want to find a random number in a different range (not between 1 and 10) then work it out like this :

RandomNumber = Int((highest no. - lowest no. + 1) * Rnd + lowest no.)

E.g. to pick a random number between 5 and 20 you would use :

RandomNumber = Int (16 * Rnd + 5)

Note - you get 16 by subtracting 5 from 20, then adding 1.

Random numbers are good for programs like lottery generators or programs involving throwing dice.

Do

Lucky Seven

In this program you will follow instructions to write a Visual Basic program which will act like a simple slot machine.

Remember there are 3 steps to writing a program on the computer:

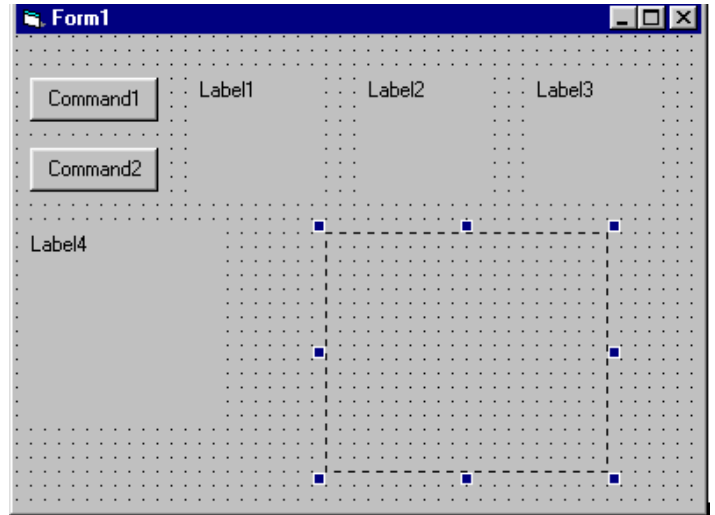
1. Place the objects onto the form
2. Change the properties of the objects
3. Write your Visual Basic code



Lucky Seven - Step 1, Put objects on Form

You will need 7 objects on your form. 2 command buttons, 4 labels and 1 image box.

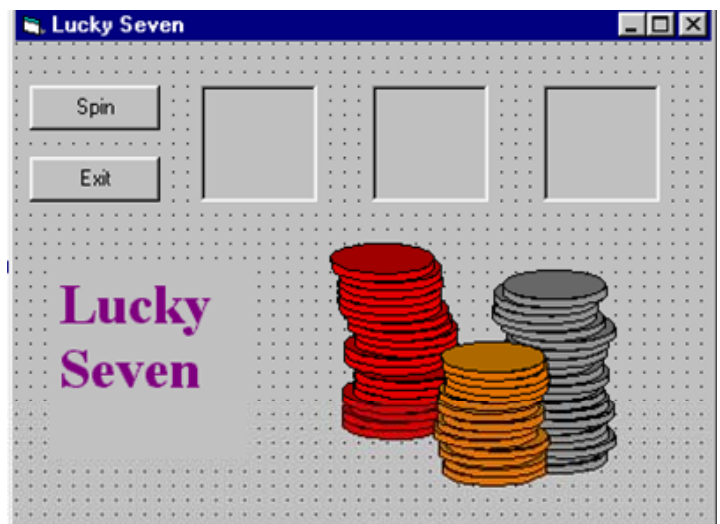
1. Open a new form
2. Add all of the objects listed above. Put them on your form like this.



You have now created your Human Computer Interface.

Lucky Seven - Step 2, change properties.

3. Change properties of all 7 objects, plus your form so that your program looks like this. (properties table overleaf)



Here are the properties:

Object	Property	Value
Form	Name	FrmLucky7
	Caption	Lucky Seven
Command Button	Name	CmdSpin
	Caption	Spin
Command Button	Name	CmdExit
	Caption	Exit
Label	Name	LblNumber1
	Caption	Blank
	BorderStyle	1
	Alignment	Center
	Font	24pt, bold
Label	Name	LblNumber2
	Caption	Blank
	BorderStyle	1
	Alignment	Center
	Font	24pt, bold
Label	Name	LblNumber3
	Caption	Blank
	BorderStyle	1
	Alignment	Center
	Font	24pt, bold
Label	Name	LblLucky7
	Font	24pt, bold, coloured
	Caption	Lucky Seven
Image Box	Name	ImgCoins
	Stretch	True
	Picture	Coins.wmf
	Visible	False

Lucky Seven - Step 3, Write Code

There are 2 command buttons which we will add code to.

Command button 1 -The Exit Button

4. In between the "Private Sub" and "End Sub" lines type the word *End*. Like this:

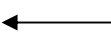
```
Private Sub CmdExit_Click()  
End 'This ends the program  
End Sub
```

The Spin Button

5. Add this code to the **cmdSpin** button after double-clicking on it:

```
Private Sub cmdSpin_Click()  
Randomize  
ImgCoins.Visible = False ' hide coins  
lblNumber1.Caption = Int(Rnd * 10) ' pick numbers  
lblNumber2.Caption = Int(Rnd * 10)  
lblNumber3.Caption = Int(Rnd * 10)  
    'if any caption is 7 display coin stack and beep  
    If (lblNumber1.Caption = 7) Or (lblNumber2.Caption  
= 7) Or  
    (lblNumber3.Caption = 7) Then  
        ImgCoins.Visible = True  
        Beep  
    End If  
End Sub
```

(Between **If**
and **Then** is
all on 1 line)



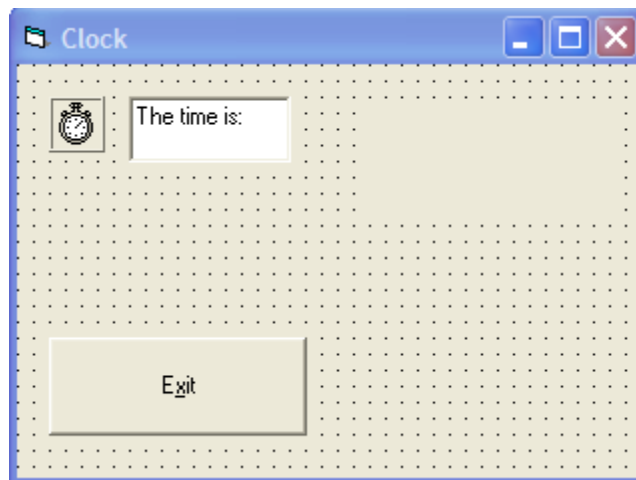
6. Now save your program, run it and test for errors.

Do

Simple Clock

In this program you will learn how to create a simple clock program, which will take the current time and display it.

1. Create a small form as shown below.



2. Change the properties as shown below:

Object	Property	Value
Form	Name	FrmClock
	StartPosition	2 Centre Screen
	Caption	Clock
Label	Name	LblTime
	Caption	(blank)
Text box	Name	txtTime
	Text	The time is:
Command Button	Name	cmdExit
	Caption	E&xit
Timer	Interval	1000

Note the ***StartPosition*** of the form. This is new to you, but makes the form always start in the middle of the screen.

3. Double-click on the **cmdExit** button and add the code to end the program.
4. Double-click on the **Timer** on your form and add the following code between Sub and End Sub.

```
Private Sub Timer1_Timer()  
    lblTime.Caption = Time  
End Sub
```

Read

Menus

Adding a menu to your program adds a 'touch of class' and makes your program even more like a Windows application. Using menus also has the advantage of clearing space on a form and generally tidying up your screen.

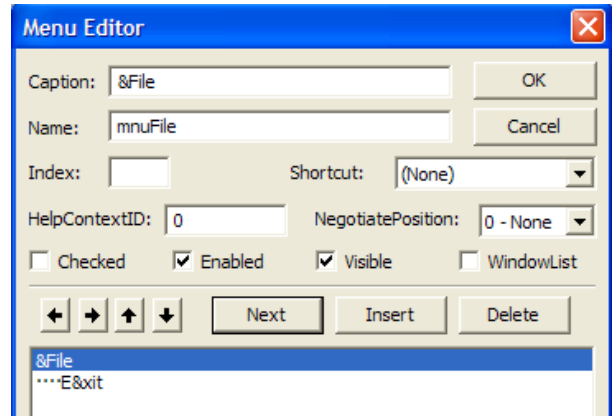
Your menus must match other Windows programs. For example, always start with a **File** menu, and write it like this - **File**, and not like this - **FILE** or **file**.

5. Click on the Tools Menu and select 'Menu Editor' or choose the Menu Editor icon.



Like many other objects, each menu must have a caption and a name. To add a File menu with Exit on it to your form, fill in the menu design box as follows :

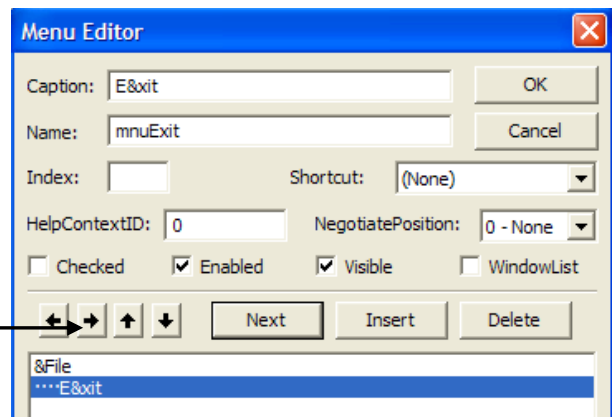
6. Complete the Menu Editor as shown with a Caption and Name for the File menu.



7. Click Next.

8. Click on the forward arrow and complete the Menu Editor with the Caption and Name for the Exit command.

Forward button



Properties for the File menu and Exit command:

Item	Caption	Name
File Menu	&File	mnuFile
Exit option	E&xit	mnuExit

We can now make our form even more compact because we have removed the need for the `cmdExit` button we have, and replaced it with a menu system.

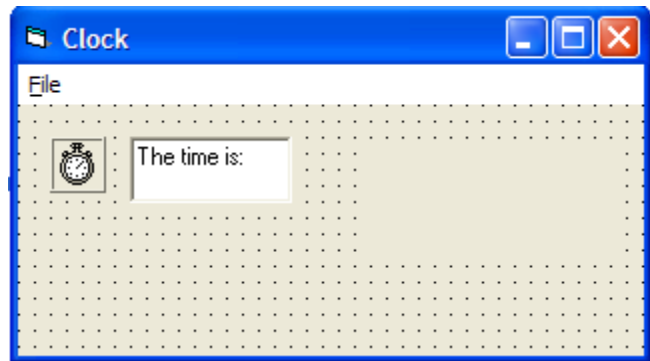
9. Choose the **Exit** command from the **File** menu.

10. Add the code to end the program to the **Exit** menu.

```
Private Sub mnuExit_Click()  
End  
End Sub
```

11. Finally delete the `cmdExit` button.

The menu design will produce this form:



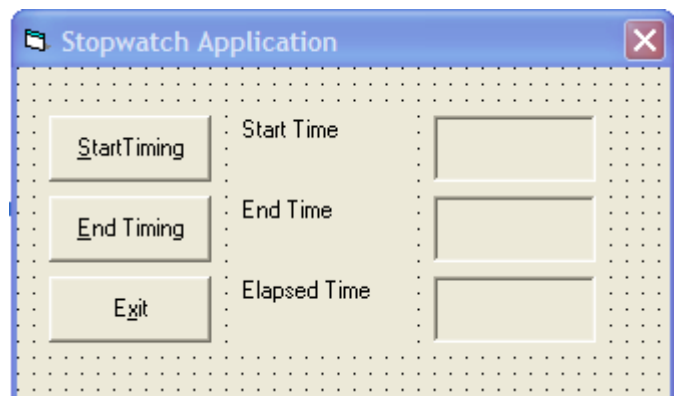
12. Save your program as *frmClock* and *prjctClock*.

Do

Stop Watch

We are going to expand the concept of our clock program and turn it into a working stopwatch. This program will allow the user to start and end the stopwatch. Then display the time that has elapsed.

1. Open the form **frmStopwatch** as shown.



2. Add the code to the **cmdExit** button.
3. Add the following code to **cmdStart** button.

```
Private Sub cmdStart_Click()  
    'Establish and print starting time  
    StartTime = Now  
    lblStart.Caption = Format(StartTime, "hh:mm:ss")  
    lblEnd.Caption = ""  
    lblElapsed.Caption = ""  
End Sub
```

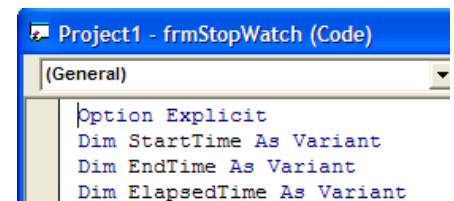
4. Add the following code to the **cmdEnd** button.

```
Private Sub cmdEnd_Click()
    'Find the ending time, compute the elapsed time
    'Put both values in label boxes
    EndTime = Now
    ElapsedTime = EndTime - StartTime
    lblEnd.Caption = Format(EndTime, "hh:mm:ss")
    lblElapsed.Caption = Format(ElapsedTime, "hh:mm:ss")
End Sub
```

5. Choose **General** from the code window drop down menu.



6. Enter the following below *Option Explicit*.



Read

We have declared variables before. These have been local variables used within an object or procedure. By placing them under general we make the variables available for use by all procedures in a form. The variant data type is a special type used by Visual Basic that can contain numeric, string, or date data. Here is a list of other variable types:

Data Type	Suffix	Data Type	Suffix
Boolean	None	Integer	%
Long (Integer)	&	Single (Floating)	!
Double (Floating)	#	Currency	@
Date	None	Object	None
String	\$	Variant	None

7. Save the changes to the form and the project.

SPL Match Statistics

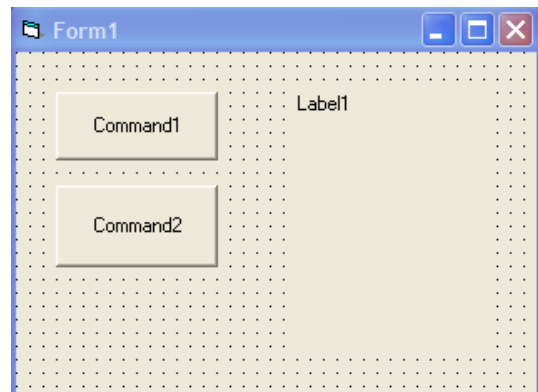
We are going to design a program that will allow a user to enter the number of goals scored in every match in the six games played each week. The program will then count the number of matches where three or more goals have been scored. The program will have two command buttons, End & Calculate and one label to display the number of games where goals scored exceed three.

Program Design

Problem (In my own words)

I have to design a program that will ask the user to enter the total goals scored in each of the SPL matches. The program will count each match where three or more goals have been scored. The program will then display the number of matches with three or more goals.

Interface Design



Property Grid

Object	Property	Value
Form	Name	frmmatchstats
	Caption	SPL Match Stats
Command Button	Name	cmdScore
	Caption	&Score
Command Button	Name	cmdExit
	Caption	E&xit
Label	Name	Lblgames
	Caption	(blank)

Algorithms for Each Button (Procedure)

Start

1. Set up variables
2. Repeat 6 times
3. Prompt user for goals scored in each match
4. Test to see if three or more goals have been scored
5. Increase count of three or more goals per match
6. Display number of matches with three or more goals

End

1. End program

Refinements

3.1 Display input box prompting user to enter total goals in a game

4.1 If goals > 3 then

5.1 variable counter = variable counter + 1

6.1 Display value of variable counter in label score

Evaluation

I found this program easy as I am quite familiar with loops, testing for conditions and using counters. All I had to do was to take these skills and adapt them to suit the program requirements. If I had to improve my program I would create a nicer user interface with graphics or coloured text and prompt the user if they didn't enter a number.

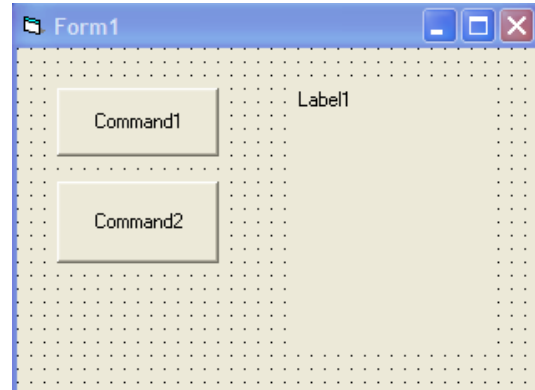
You now, are going to create the program above from design through to evaluation using the sheets provided.

1. Copy the problem statement, design grid and properties table shown above.
2. Create the HCI, change the object properties and add the code.

Show your teacher your design sheets.

Do

3. Create your form with two command buttons and one label. As shown.



4. Change the properties of each object including the form as per your design.
5. Double-click on the **cmdExit** command and add the code to end the program.
6. Double-click on the **cmdScore** button and add the following code.

```
Private Sub cmdScore_Click()  
Dim matches, goals As Integer  
For loops = 1 To 6  
goals = InputBox("please enter number of goals")  
If goals >= 3 Then matches = matches + 1  
Next  
lblgames.Caption = "total games with 3 or more goals= " & matches  
  
End Sub
```

Note: Change the label names to suit the properties of **your** labels.

7. Save your project and form with a suitable name.
8. Run your program to see if it works.
9. Demonstrate your working program to your teacher.

Copy the Over Forty assessment from the PrepWork folder into your own folder.

PRACTICAL ABILITIES TASK

OVER FORTY

Your teacher will outline how you should do this task and give you opportunities for discussion. Read and follow all instructions carefully, use hardware and software properly and write your report neatly.

You are required to write a program to count the number of pupils in a class of twelve who score over 40 in a test out of 60. The pupils' marks are to be entered at the keyboard. The output is a message that indicates the number of pupils who score more than 40.

1 DESCRIBE METHOD - Analysis *3 marks*

Show that you understand what is required by describing how you will do this task.

2 LIST STEPS - Design *6 marks*

On a separate sheet of paper, show the main steps and refinements.

3 ENTER PROGRAM - Implementation *5 marks*

Enter the program listing. Include internal commentary. Correct any mistakes that you make. Save the listing.

4 TEST PROGRAM - Testing *5 marks*

Complete the table below, firstly by writing three more sets of test data under the twelve marks listed. The given marks are all above 40. The next set should all be below 40. The next set should have marks above, below and including 40. The last set should only have marks 39, 40 and 41. Complete the second column

by counting and writing down how many marks in each set are over 40. Complete the last column by recording the program's output with your test data.

pupils' marks	my count over 40	program's count over 40
46, 54, 51, 41, 49, 58, 45, 44, 53, 50, 57, 43		

Compare your count of marks over 40 with the program's output and write an appropriate comment.

5 GET PRINTOUTS - Implementation

2 marks

Get **two** printouts with your name in the footer. One should show your listing and the other a run using **one** of the sets of test data above.

6 SUGGEST IMPROVEMENT - Evaluation

2 marks

Write down **one** way in which your program could be better.

7 JUDGE PERFORMANCE - Evaluation

2 marks

Read this task sheet again. Describe how well you think you have done the task.

Grade 3 - 18 or more

Grade 4 - 17 to 13

Grade 5 - 12

Grade 7 - 11 or less