

## **Module 3.6 Programming II**

In Module 3.3 you learned the basics of creating a Visual Basic program. You learned the fundamentals of adding controls to a form and attaching code to these controls.

You were able to input values using an input box or text box, perform simple calculations and output values to a label.

However, you have learned only a fraction of the commands available in Visual Basic. In this module you will learn how to select between different sections of code and how to execute sections of code more than once.

## Task 1 – IF Statements

Sometimes it is necessary to carry an action if a condition is met. For example:

If homework is not completed Then give pupil 20 PLs

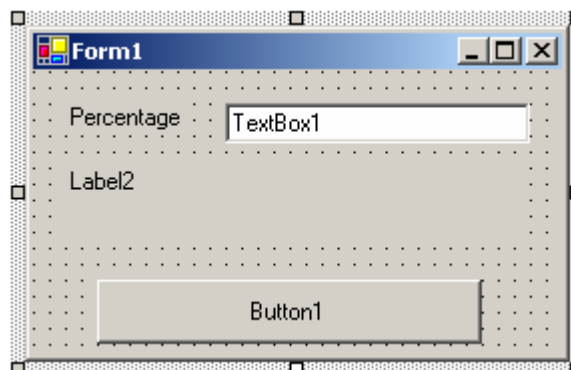
In Visual Basic the following syntax is used:

```
If [condition] Then [code]
```

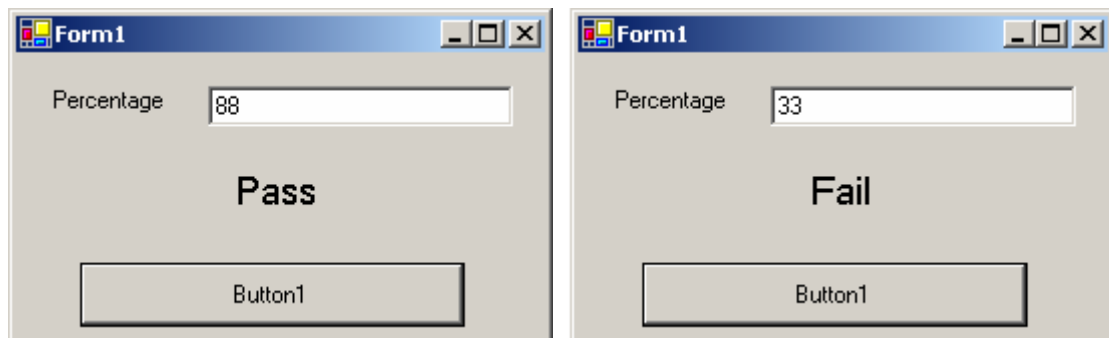
Sometimes you want to choose between different options. In this case you must use the following syntax:

```
If [condition] Then  
    [code]  
Else  
    [code]  
End If
```

- Create a new application called If Statements
- Create the form below:



- Code the button so that, when the user enters an exam percentage in the text box, the word Pass or Fail is displayed on the label.



- Build and run this application.

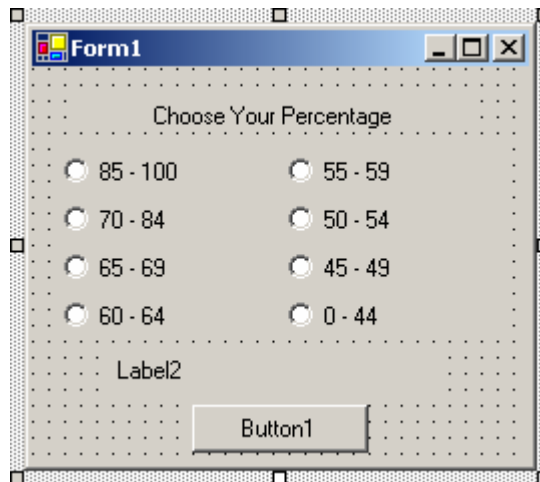
## Task 2 - Radio Buttons

The If statement used in Task 1 is used frequently in Visual Basic. One of the objects that you can place on a form and that makes use of If statements is a radio button.

You can choose to different tasks depending on which radio button is selected. The VB code is shown below.

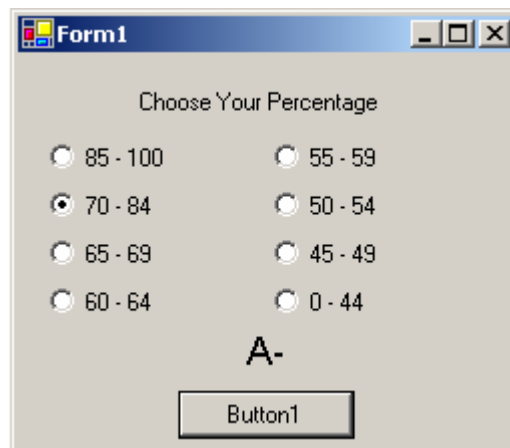
```
If RadioButton1.Checked Then  
    [code]  
End If
```

- Create a new application called Radio Buttons.
- Create the form below:



- Code the button so that, when a radio button is selected and the button pressed, the program will display the grade for the selected percentage as shown below. Use the condition:

If RadioButton1.checked Then put A- into the label  
etc.



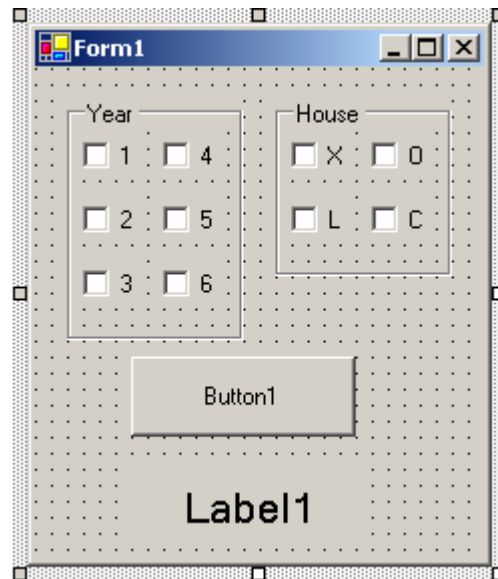
- Build and run this application.

### Task 3 - Check Boxes

Check boxes are used in a similar way to radio buttons but are different in that none, some or all check boxes can be selected at one time.

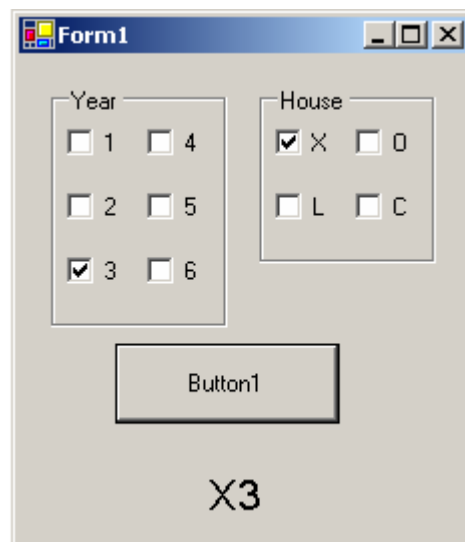
```
If CheckBox1.Checked Then  
    [code]  
End If
```

- Create a new application called Check Boxes
- Create the form below, use a Group Box to surround the groups of check boxes.



- Code the button so that, when the button is pressed, the house and year of a pupil is displayed in the label as shown below. Use the following code:

```
If CheckBox3.checked Then Year = 1 etc.  
If CheckBox7.checked Then House = X etc.  
Output House & Year to label
```



- Build and run this application.

## Task 4 – Select Case Statements

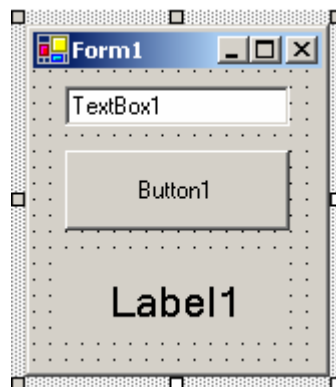
Consider task 1 where you entered an exam percentage and outputted the word Pass or Fail. This program could be altered to return the grade for each percentage. Using If statements you would have very complicated code e.g.

If mark > 85 then grade A+ else If mark > 70 then grade = A- else If mark > 65 then grade = B+ etc.

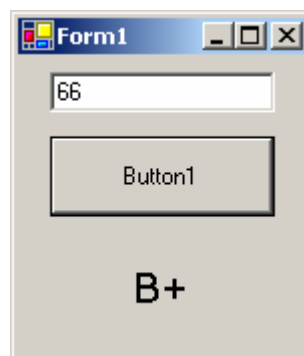
A better solution is to use a Select Case statement. The syntax is shown below:

```
Select Case [test expression]
Case [condition 1]
    [action 1]
Case [condition 2]
    [action 2]
Case [condition 3]
    [action 3]
Case Else
    [action 4]
End Select
```

- Create a new application called Select Case Statements
- Create the following form:

A screenshot of a Windows application window titled 'Form1'. The window contains three controls: a text box at the top labeled 'TextBox1', a button in the middle labeled 'Button1', and a label at the bottom labeled 'Label1'. The window has a standard Windows title bar with minimize, maximize, and close buttons.

- Code the button so that, when the user enters an exam percentage and presses the button, the grade corresponding to the percentage is displayed.
- In this case the Test expression is the variable containing the exam result and the conditions are 85 to 100, 70 to 84 etc.

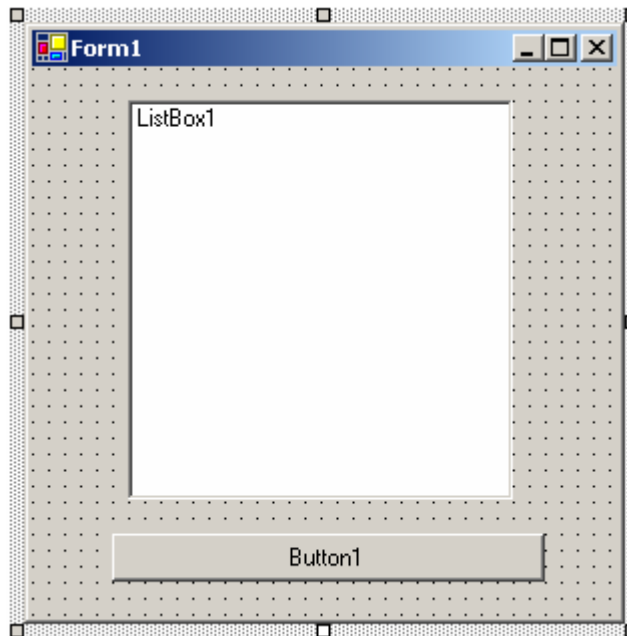
A screenshot of the same application window 'Form1'. The text box now contains the number '66'. The button is still labeled 'Button1', and the label below it now displays the grade 'B+'.

- Build and run this application

## Task 5 – List Boxes

List boxes are used frequently in Visual Basic, particularly when using loops to store results of repeated code.

Create the following form.



To output a single item to a listbox:

```
ListBox1.Items.Add("Hello")
```

To remove

```
ListBox1.Items.Remove("Hello")
```

Practice using these techniques by creating a solution to the following problem.

- Write a program that, using input boxes, prompts the user for 5 names and adds these names to a list box.
- Carry out the following steps five times:
  1. Prompt for name
  2. Add name to list

## Task 6 – Unconditional Loops

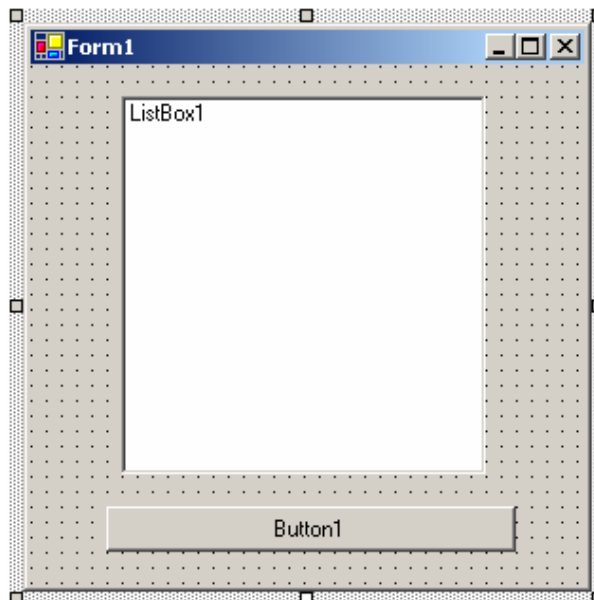
Often you want to carry out instructions several times, e.g. calculate the percentage for each student in a class. In the previous task you had to carry out the same operations 5 times – displaying an input box 5 times, entering the 5 names and adding the 5 names to a list box 5 times. By using an unconditional loop you can enter the instruction once and tell the computer to carry it out 5 times.

In Visual Basic the For..Next loop is used to carry out the instructions a fixed number of times. This loop has the following syntax:

```
For counter = start To end
    [statements]
Next counter
```

Counter is a control variable, used to count the number of times the loop is executed.

- Create the following form



- Attach the following code to button1

```
Dim counter As Integer
For counter = 1 To 10
    ListBox1.Items.Add("Item" & counter)
Next
```

Practice using these techniques by creating a solution to the following problem.

- Write a program using an unconditional loop that, using input boxes, prompts the user for 5 names and adds these names to a list box.
- Use the following algorithm:
  1. For counter = 1 to 5
  2. Prompt for name
  3. Add name to list
  4. End loop

## Task 7 – Conditional Loops

The second type of loop you will use are conditional loops.

Sometimes the condition is at the start of the loop, in this way the loop is always executed at least once.

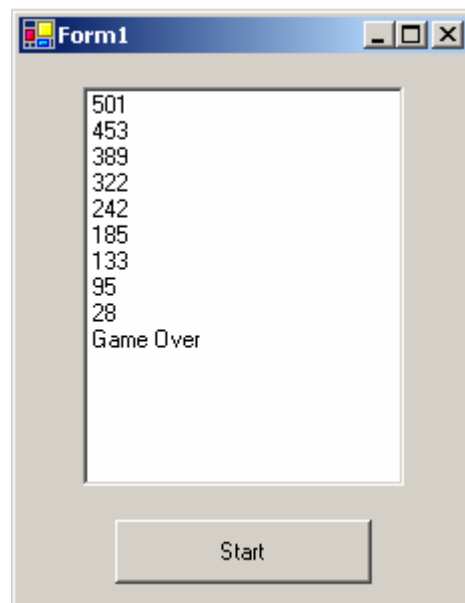
```
Do  
  [code]  
Loop While/Until Condition
```

Sometimes the condition is at the end of the loop, in this way the loop may not be executed.

```
Do While/Until Condition  
  [code]  
Loop
```

Practice using these techniques by creating a solution to the following problem.

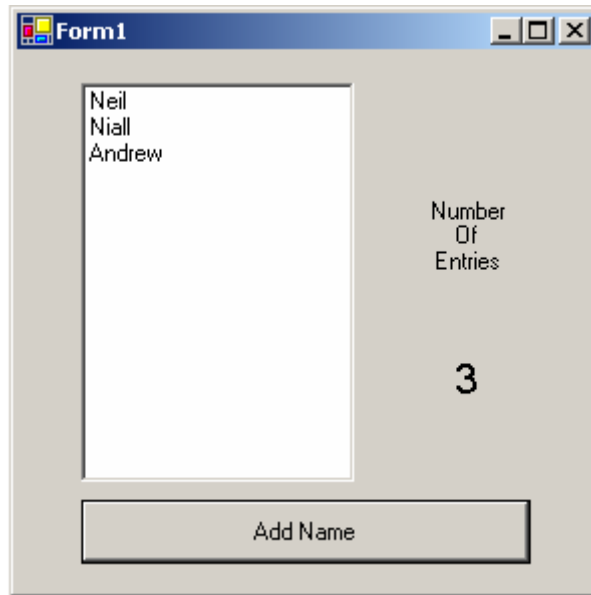
- You must create a program that can be used to calculate dart scores. The number, displayed in a list box, should start at 501. Then, when you press the start button, the program will repeatedly ask for the dart scores until the total is zero or less. The program should look like the following:



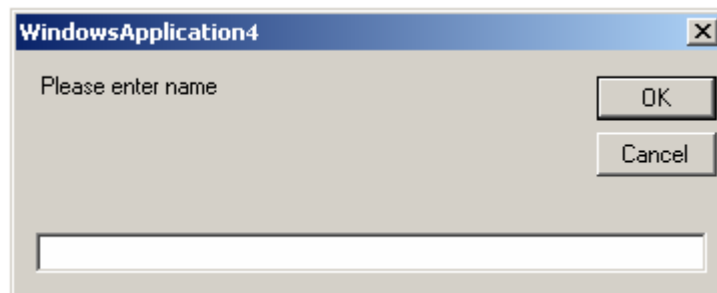
- Use the following algorithm:
  - Set total score to 501
  - Loop
  - Get score
  - Subtract score from total score
  - Loop until score is zero or less

## Task 8 – Loop and Count

- Create the following form using one button, one list box and two labels.



- When the button is clicked the following Input Box should appear.



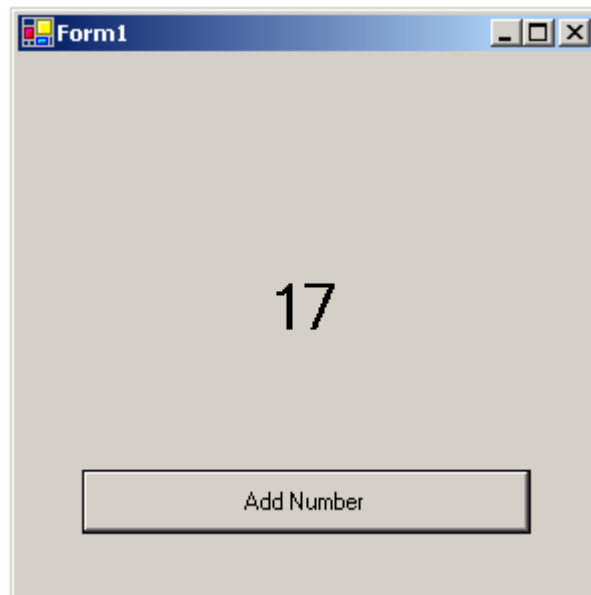
Entering a name and pressing OK should add that name to the list box and also increment (add one to) the number of entries. This can be achieved by having a counter variable and adding one to the variable each time the loop is executed.

Entering the word End should terminate the loop. End should not be counted as an entry to the list.

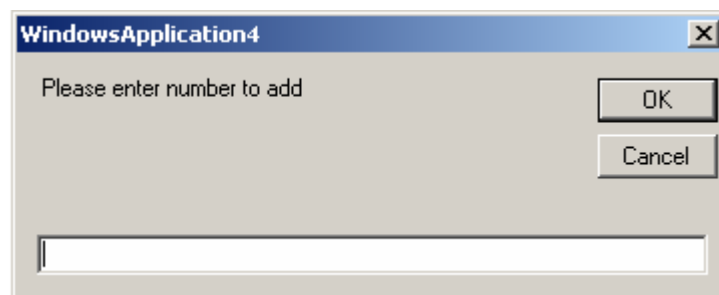
- Use the following algorithm:
  1. prompt for name
  2. loop until name = end
  3. add name to list
  4. add one to counter
  5. display counter
  6. prompt for name
  7. end loop

## Task 9 – Loop and Add

Create a form with one button and one label



Pressing the button should bring up the following Input Box.



As you enter numbers in this input box each number is added to a running total and displayed in the label.

This loop should continue until zero is entered in the input box.

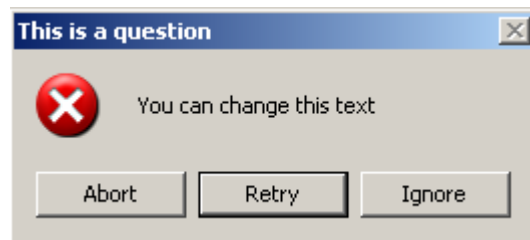
- Use the following algorithm
  1. Loop
  2. Prompt for number
  3. Add number to total
  4. Display total
  5. Loop until number entered = zero

## Task 10 – Message Boxes

Message boxes are dialog boxes that pop up to warn you or to ask you to confirm an action. You can change all properties of a message box.

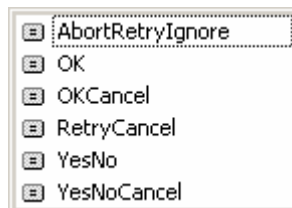
The message box is summoned using the following code, entered all on one line:

```
MessageBox.Show("Are You Sure", "Title Box",_  
MessageBoxButtons.YesNoCancel, MessageBoxIcon.Asterisk,_  
MessageBoxDefaultButton.Button2)
```

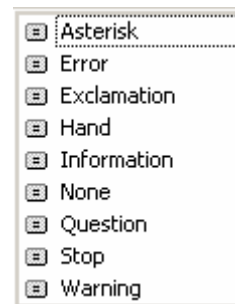


The caption on the message box and in the title bar of the box can be changed and which button is set as the default can also be changed.

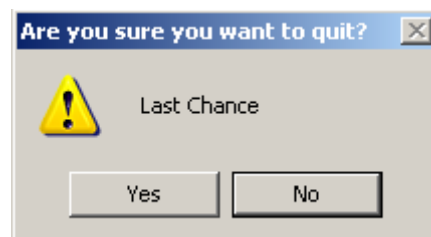
The buttons displayed can be any combination of the following:



MessageBoxIcon can be chosen from the following:



- Write code to produce the following message box.



- Add the following code to your program so that, when the Yes button is selected, the program will close. Use the following code

```
Dim result As DialogResult  
result = [message box code]  
If result = DialogResult.Yes Then  
Me.Close()  
End If
```

## Do either Assessable Task 1 OR Assessable Task 2

### Assessable Task 1

In Ghana everyone is given a name depending on the day of the week on which they were born.

	<b>Male</b>	<b>Female</b>
<b>Sunday</b>	Kwesi	Akosua
<b>Monday</b>	Kojo	Adjoa
<b>Tuesday</b>	Kwabena	Abena
<b>Wednesday</b>	Kweku	Akua
<b>Thursday</b>	Yaw	Yaa
<b>Friday</b>	Kofi	Efua
<b>Saturday</b>	Kwame	Ama

A program is required that allows the user to enter their gender and their birth day. The program then outputs the user's Ghanaian name.

The program may be implemented using buttons, menus, check boxes, radio buttons, text boxes, input boxes or any other appropriate Visual Basic object.

Produce and submit the following documentation:

#### **Analysis**

A requirements specification stating clearly what the program must do and a list of all inputs and outputs with their data types.

#### **Design**

A sketch of the forms, dialog boxes, input boxes, message boxes etc. that your program will display. An algorithm or structure diagram showing the logic of the program.

#### **Implementation**

A printout of your VB code and forms.

#### **Testing**

An expected and actual test table with normal, extreme and exceptional test data.

#### **Documentation**

A brief user guide and technical guide.

#### **Maintenance**

A maintenance document detailing any corrective, perfective or adaptive maintenance you have undertaken.

#### **Evaluation**

A response to the evaluation questions to evaluate each stage of the software development process.

## **Assessable Task 2**

A program is required that prompts the user to enter a number between one and ten and output that number's times table. If the user enters a number outwith the range an error message should be displayed.

The program should use a conditional loop to ensure the number is within the valid range and an unconditional loop should be used to output the times table.

Produce and submit the following documentation:

### **Analysis**

A requirements specification stating clearly what the program must do and a list of all inputs and outputs with their data types.

### **Design**

A sketch of the forms, dialog boxes, input boxes, message boxes etc. that your program will display. An algorithm or structure diagram showing the logic of the program.

### **Implementation**

A printout of your VB code and forms.

### **Testing**

An expected and actual test table with normal, extreme and exceptional test data.

### **Documentation**

A brief user guide and technical guide.

### **Maintenance**

A maintenance document detailing any corrective, perfective or adaptive maintenance you have undertaken.

### **Evaluation**

A response to the evaluation questions to evaluate each stage of the software development process.

## Support Tasks

These tasks should be used as further practice in the fundamental aspects of programming. Implement these programs in Visual Basic.

1. People may serve on jury duty if they are between the ages of 18 and 65. Write a program that allows the user to enter their age and the program will state whether they are eligible for jury duty or not.
2. A salesperson receives a bonus dependent upon sales. Sales of £100,000 or more are rewarded with a £10,000 bonus. Sales of £70,000 to £99,999 are rewarded with a bonus of £7,000 and if the sales are between £50,000 and £69,999 the bonus is £4,000. Sales of less than £50,000 do not receive a bonus. Write a program that allows a salesman to enter their sales and uses a Select Case statement to calculate their bonus.
3. Ask the user to input two different numbers. Use a FOR/NEXT loop to print all the numbers between the two values they enter.

## Extension Tasks

These tasks should be used as further practice in more complicated tasks that can be used to advance your skills in implementing programs to solve a variety of problems. In each problem take the opportunity to use a variety of objects from the toolbox.

1. A sports club has three categories of membership charge. Juniors (aged up to 18) pay £60 per year, Seniors (19-49) pay £120 and Veterans (50 and over) pay £80. Juniors who have been a member for 2 years or more get a £20 reduction. Seniors and veterans who have been members for 10 years or more get a £30 reduction. Write a program that asks for a member's age and the number of years they have been a member, and outputs their category of membership and how much they must pay.
2. Allow the user to enter as many positive whole numbers as they wish and to enter 0 to indicate they have finished. Then display the number of even values and number of odd values entered by the user. Use the **Mod** operator to work out whether a number is odd or even.
3. Write a program that allows the user to enter a number. In a Rich Text Box you should display as many choruses necessary of the Green Bottles song, e.g. 5 green bottles hanging on a wall etc.

## Homework Exercise 1 – Selection

1. The following program is executed.

```
Dim age As Byte
age = InputBox("Enter your age")
If age < 17 Then
    Label1.text = "Too young to drive"
Else
    Label1.text = "Eligible to drive"
End If
```

- (a) What is the effect of line 1? (1)
- (b) What is the effect of line 2? (1)
- (c) What is the effect of lines 3 to 7? (3)

2. The following program is executed.

```
Dim age As Byte
age = InputBox("Enter your age")
Select Case age
    Case 0 To 17
        Label1.text = "Too young for jury duty"
    Case 18 To 65
        Label1.text = "Eligible for jury duty"
    Case 65 To 120
        Label1.text = "Too old for jury duty"
    Case Else
        Label1.text = "Invalid age entered"
End Select
```

- (a) What is the effect of line 3? (1)
- (b) What is the effect of lines 6 and 7? (2)
- (c) What is the effect of lines 10 and 11? (2)

Total (10)

## Homework Exercise 2 – Iteration

1. The following program is executed.

```
Dim number1, counter, total As Byte
number1 = InputBox("Enter Number")
total = 0
For counter = 1 To number1
    total = total + 2
Next
label1.text = total
```

- (a) What would happen if, when prompted, the user entered 1?  
Explain your answer. (2)
- (b) What would happen if, when prompted, the user entered 5?  
Explain your answer. (2)

2. Consider the following program.

```
Dim number1 As Byte
Do While number1 <> 3
    number1 = InputBox("Guess The Lucky Number")
Loop
Label1.Text = "3 is the magic number"
```

- (a) What would happen if, when prompted, the user entered 2?  
Explain your answer. (2)
- (b) What would happen if, when prompted, the user entered 3?  
Explain your answer. (2)

3. Is the loop in Question 1 a conditional or unconditional loop? (1)
4. Is the loop in Question 2 a conditional or unconditional loop? (1)

Total (10)

### Homework Exercise 3

1. An unconditional loop is executed a fixed number of times. What is the syntax of an unconditional loop in Visual Basic? (1)
2. A conditional loop is executed an unspecified number of times. Write the syntax of two alternative types of conditional loop. (2)
3. Write the syntax of a simple decision. (1)
4. Write the syntax of a complex decision. (2)
5. Write the syntax of a multiway selection. (1)

6. Consider the following code:

```
Dim number, total As Short
total = 1
Do
    number = InputBox("Enter number")
    total = total * number
Loop Until (number = 1)
Label1.Text = total
```

What would appear in Label1 if the following numbers were entered?

- (a) 2,3,1 (1)
- (b) 5,0,1 (1)
- (c) -1,-2,3,-3,1 (1)

Total (10)

## Homework Exercise 4

1. Consider the following code:

```
Dim number, total As Short
total = 1
number = InputBox("Enter number")
Do Until number = 1
    total = total * number
    number = InputBox("Enter number")
Loop
Label1.Text = total
```

What would appear in Label1 if the following numbers were entered?

- (a) 2, 3 1 (1)
- (b) 1, 2, 3, 1 (1)
- (c) -1, -2, -3, 1 (1)
- (d) Why does the loop use 1 as the termination value, i.e. why use 1 as the number that stops the loop? (1)
- (e) Why would 0 be an inappropriate termination value? (1)

2. Consider the following code:

```
Dim counter, number, total As Short
total = 0
For counter = 1 To 5
    number = InputBox("Enter number")
    If number > 10 Then
        total = total + number
    Else
        total = total - number
    End If
Next
Label1.Text = total
```

What would appear in Label1 if the following numbers were entered?

- (a) 5,10,15,20,25 (1)
- (b) -10,10,25,0,5 (1)

3. Consider the following code:

```
Dim counter, number, total As Short
total = 1
For counter = 1 To 5
    number = InputBox("Enter number")
    If number > total Then
        total = total * number
    Else
        total = total / number
    End If
Next
Label1.Text = total
```

What would appear in Label1 if the following numbers were entered?

(a) 5, 6, 3, 5, 4 (1)

(b) 10,-2,-1, 2, -2 (1)

(c) Why could the following values cause the program to crash?  
3,6,9,0,12 (1)

Total (10)

## Homework Exercise 5

1. Explain the meaning of the following:
    - (a) Number +=1 (1)
    - (b) If number <> 5 then (1)
  
  2. Convert the following to Visual Basic code.
    - (a) Set the value of the variable called Total to zero. (1)
    - (b) Add 1 onto the number stored in the variable called total. (1)
    - (c) If the number stored in the variable total is equal to 0 then write 'The program is finished' inside label1 on the form. (2)
    - (d) Declare three variables called number, product and counter to store small, whole numbers. Set product to equal 1 at the start of the program. Use an unconditional loop to ask for five numbers and multiply these numbers together, storing the result in the variable called product. Display the final result in label1. (3)
    - (e) Declare a variable called Guess to store a small, whole number. Use a loop to repeatedly ask for a number. If the number doesn't equal 7 the program should say 'Lucky number not found' until 7 is entered then a congratulatory message should be displayed in a label. (3)
- Total (12)