Introduction to Programming

SAMPLE

Examination Paper

Answer ALL questions.

Produce a Word document using the instructions provided in the 'Background information' section.

Time: 3 hours

The maximum mark for this paper is 100.

Any reference material brought into the examination room must be handed to the invigilator before the start of the examination.

Use of any PC equipment other than that provided in the examination room is <u>not</u> permitted

<u>A formula sheet is provided at the end of the question paper.</u>

Complete the following computer programming tasks using Visual Basic.

For each task you must test the programming code you create.

The evidence you **must produce is a Word document** containing the following for each task.
- Required outcomes
- Simple test criteria
- The Visual Basic program code, with relevant comments (i.e. annotated)
- Screenshots of the user interface
- Screenshots of the output generated when the program code is run

It would be a good idea to use the Windows Snipping tool to assemble this Word document.

You do **not** have to provide any general commentary on your work.
You do **not** have to provide any history of what debugging you may have done.

It is very important that the program code is clear, the outputs are designed to be easy to understand and your Word document is well presented.

A formula sheet containing the VB syntax for common programming structures is provided at the back of the examination paper and you may refer to this during the exam. Use of your own notes during the examination is not allowed.

Marks

Question 1

Write and test a program to convert acres to hectares. 1 acre = 0.404686 hectares. **10**
The program should accept user input via a textbox and display the output using a
label. The program should work for real numbers as well as whole numbers (e.g. it
should accept 7.2 acres, 0.5 acres, etc.).

Total 10 Marks

Question 2

Write and test a program that allows an athletics trainer to store the names of athletes **50**
and the maximum distance they can throw the javelin in metres, to three decimal
places. The athletics trainer needs to store data about 16 athletes.
  - This data should be stored in two separate arrays (one array for the athlete's
    names and another array for maximum distance).
  - The athletes' names should be stored as strings.
  - The maximum distances should be stored as real numbers.
  - The input data should be validated before being entered into the array. Athlete
    names must be at least 2 characters long and cannot contain more than 35
    characters. Only values between 50 and 130 metres should be accepted for
    the maximum distance.
  - If an invalid entry is attempted, a message should be displayed informing the
    user that the entry will not be stored and that the athlete name must be
    between 2 and 35 characters long and that only values between 50 and 130
    will be accepted for the maximum distance.
  - Once the arrays are full, the 'Input data' button should be disabled and the
    'Show contents' button should be enabled which, when pressed, will print out
    the athlete name and maximum distance for each athlete, in the order entered,
    to a label.
  - Complete an Object Definition Sheet for this task.

Total 50 Marks

Question 3

The athletics trainer wishes to find the names of all athletes whose maximum javelin **15** throwing distance is at least 90 metres in order to invite these promising javelin throwers to participate in a more intensive training regime with the aim of entering them in national competitions.

Extend the program from Question 2 so that it includes a button which, when pressed, will identify the names of all athletes with a maximum javelin throwing distance of 90 metres or higher. The program should store, in a new array, the names of all the athletes who meet the throwing distance criteria, so that they can be contacted. The list of promising athletes should be displayed on a label.

Total 15 Marks

Question 4

The athletics trainer wants a search facility added to the program to enable them to **10** search for an athlete and find their maximum javelin throw distance. Extend the program from Question 3 to add a textbox to enter an athlete name, a button which, when pressed, will run the search, and a label to display their maximum distance.

Total 10 Marks

Question 5

The athlete names and maximum distances for all athletes need to be stored **15** permanently. Extend the program from Question 4 so that it writes out the athlete name and corresponding maximum distance. These should be written out to a text file called 'JavelinDistance'.
- The athlete names and their maximum distances should be paired on the same line, with the first athlete name followed by their maximum distance on the same line with no spaces between them, and so on for all athletes in the array.
- There should be a 'Save results' button, which will write the data to file when pressed. This button should not be enabled until the 'Find promising javelin throwers' button (or whatever name was given to the button for Question 3) has been pressed.
- Test the extended functionality of the program.

Total 15 Marks

Once you have completed the tasks, carefully check the contents of the Word file so that it is a complete record of your work and is presented according to the guidance notes above.

Submit this Word file for assessment via suitable media.

End of paper

Formula sheet

Declaring variables

Variable declarations have the following format:

Dim [variable_name] As [data-type]

Code examples

```
Dim myWholeNumber As Integer
Dim myRealNumber as Double
Dim myName As String
```

For loops

For loops have the following format:

Dim [loop_counter_name] As Integer

```
For [loop_counter_name] = [start_num] To [end_num]
    'do the code here
Next
```

Code example

```
Dim counter As Integer

For counter = 0 To 5
    Console.WriteLine("The counter is at: " & counter)
Next
```

While loops

While loops have the following format:

```
While [condition]
    'do the code here
Next
```

Code example

```
Dim myNum As Integer

While myNum < 12
    Console.WriteLine("The value of myNum is: " & myNum)
    myNum=myNum+1
End While
```

If statements

If statements have the following format:

If [condition]
    'do the code here
End If

Code example

```
Dim myNum As Integer

If myNum < 12
    Console.WriteLine("The value of myNum is: " & myNum)
End While
```

Arrays

Arrays are declared and the number of elements set as follows:

Dim [array_name]([number_of_elements]) As [data_type]

Code example

```
Dim myNums(12) As Integer
```

To store a value in an array we need to indicate which element we want to store the value in:

```
myNums[0] = 15            'stores the value 15 in element 0 of the array
```

To retrieve a value from an array we need to indicate which element we wish to access:

```
temp = myNums[2]          'copy contents of element 2 of myNums to temp
```

To loop over an array using a for loop:

```
Dim counter As Integer
For counter = 0 to 11
    temp = myNums[counter]
    Console.WriteLine("The counter is at: " & counter)
Next
```