



INTRODUCTION TO PROGRAMMING

SEPTEMBER 2013

Examination Paper

Answer ALL questions.

Clearly cross out surplus answers.

Time: 3 hours

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

A formula sheet is provided at the end of the examination paper.

PLEASE DO NOT WRITE ON THIS EXAMINATION PAPER

**ALL ANSWERS SHOULD BE PRODUCED ON YOUR PC
AND PRINTED AS DIRECTED IN THE EXAMINATION PAPER**

Produce a Word document following the instructions given in *Background Information* below.

Use of any PC equipment other than that provided in the examination room is NOT permitted.

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.

Background Information

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)
- Screenshot or screenshots of the user interface
- Screenshot or 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 end of the examination paper

Marks

TASKS

1. Write and test a program to convert miles into kilometres. 1 miles = 1.61 kilometres. 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 2.5 miles, 7.2 miles etc). **10**

2. Write and test a program that allows the user to enter ten whole numbers, one at a time, and store these in an array. The input data should be validated before entering it into the array. Only values between 0 and 100 should be accepted and stored in the array. If an invalid entry is attempted, a message should be displayed informing the user that the number will not be stored and that a value between 0 and 100 must be entered. Once the array is full, the “Input number” button should be disabled and the “Show contents” button should be enabled which, when pressed, will print out the contents of the array, in the order entered, to a label. Complete an Object Definition Sheet for this task. **40**

3. Extend the program from task 2 so that it will find and display the following: **38**
- the highest number in the array
 - the lowest number in the array
 - the mean (average) value of the numbers in the array
- Test the extended functionality of the program.
4. The user wants the option of storing the contents of the array in a text file. Extend the program from Tasks 2 and 3 by providing a write to file option that will write the contents to a text file when a button is clicked. The user should provide the name of the file via a textbox. Show evidence that the file has been created. **12**

Total 100 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 presented according to the guidance notes above.

Submit this Word file for assessment via suitable media.

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
```