# Topic: 2.2.2 Data structures; arrays

**Define and use arrays (one- and two-dimensional) for solving simple problems (this should include initializing arrays, reading data into arrays and performing a simple serial search on a one-dimensional array)**

A data structure is a collection of different data items that are stored together in a clearly defined way. Two common data structures are arrays and records.

## Array

An array is a data structure, which allows a set of items of identical data type to be stored together using the same identifier name.

Arrays are declared in a similar way to standard variables, except that the array size and dimensions are included. For example, the following declares an array that reserves five locations in memory and labels these as ‘Names’:

```
DIM Names(4) As String
```

The five individual locations are Names(0), Names(1), Names(2), Names(3), Names(4).

Each data item is called an element of the array. To reference a particular element a programmer must use the appropriate index. For example, the following statement assigns data to the 5th element:

```
Names(4) = “John”
```

Arrays simplify the processing of similar data. An algorithm for getting five names from the user and storing them in the array Names is shown below:

```
Dim Names(4) As String
Dim name As String

For i=0 to 4

      Input name

      Names(i)=name

Next i
```

## Topic: 2.2.2 Data structures; arrays

**One-dimensional arrays**

A one-dimensional array is a data structure in which the array is declared using a single index and can be visually represented as a list.

The following diagram shows the visual representation of the array Names(4):

| Index | Element |
|---|---|
| [0] | JONES |
| [1] | ERICSON |
| [2] | WILLIAMS |
| [3] | ADAMS |
| [4] | JOHAL |

**Questions:**

**Write and test a program to complete the three tasks.**

**TASK 1**

Input and store the names and marks for 30 students who have sat three computer science tests. Test 1 is out of 20 marks, Test 2 is out of 25 marks, Test 3 is out of 35 marks. You must store the names in a one-dimensional array and the marks and total score for each student in one-dimensional arrays. All the marks must be validated on entry and any invalid marks rejected. You may assume that the students' names are unique.

**TASK 2**

Calculate and store the total score for each student and calculate the average score for the whole class. Output each student's name followed by their total score, then output the average score for the class.

**TASK 3**

Select the student with the highest score and output their name and score.

Your program must include appropriate prompts for the entry of data. Error messages and other output need to be set out clearly and understandably. All variables, constants and other identifiers must have meaningful names. Each task must be fully tested.