

# AQA Computer Science AS-Level

## 3.2.1 Data structures and abstract data types

Advanced Notes



## **Specification:**

### **3.2.1.1 Data structures:**

Be familiar with the concept of data structures.

### **3.2.1.2 Single- and multi-dimensional arrays (or equivalent):**

Use arrays (or equivalent) in the design of solutions to simple problems.

### **3.2.1.3 Fields, records and files:**

Be able to read/write from/to a text file.

Be able to read/write data from/to a binary (non-text) file.



## Data structures

Data structures are used by computers as the **containers** within which information is stored. Different data structures exist and some are better suited to **different types of data** than others. When storing data, a programmer must decide which of the data structures available is the best to use.

### Arrays

An array is an **indexed set of related elements**. An array must be **finite**, **indexed** and must only contain elements with the same **data type**.

Array Names = { "George", "Sue", "Mo" }

The elements of an array are given an **index**, which often **starts from zero**. For example, with the array shown above, Names(2) would return "Mo" as the first item ("George") is given the index 0.

The array shown above is a **one-dimensional array** which could be visualised with the following table:

0	1	2
"George"	"Sue"	"Mo"

Arrays can be created in **many dimensions**. For example, a two-dimensional array could look like this:

Array Maze = { {Wall, Path, Wall}, {Path, Path, Wall}, {Wall, Path, Wall} }

When displayed in a **table**, the Maze array starts to make a little more sense:

	0	1	2
0	Wall	Path	Wall
1	Path	Path	Wall
2	Wall	Path	Wall

When an individual element is referenced, the **x coordinate is listed first**.

For example, Maze(1, 2) would return Path and Maze(2, 1) would return Wall.

#### Synoptic Link

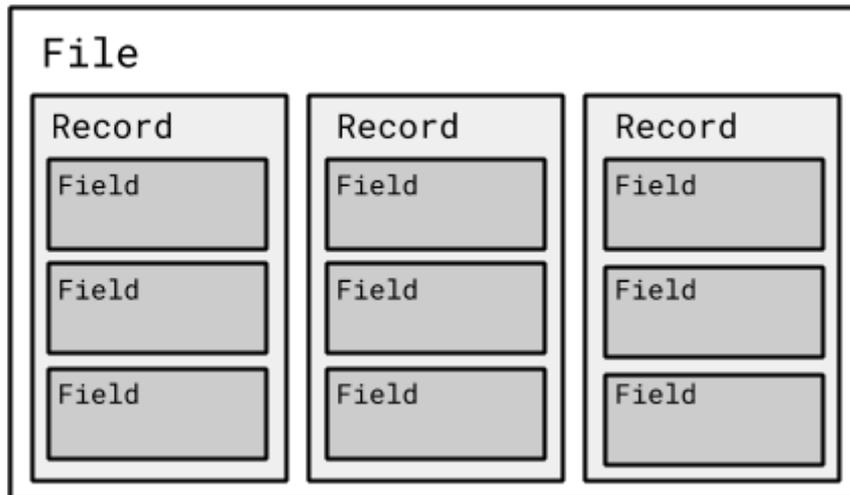
A **data type** is defined by the values it can take or the operations which can be performed on it.

Data types are covered in the notes for **fundamentals of programming**.



## Fields, records and files

Information is stored by computers as a **series of files**. Each file is made up of **records** which are composed of a number of **fields**.



It's important that you make sure you can write to and read from files in your chosen programming language.

