January 1, 2018

Chris Dworczyk

WLC

Master notes

Data Structures

Contents

[Data Structures Introduction 2](#_Toc503700011)

[What is Data? 2](#_Toc503700012)

[Data types – pascal 2](#_Toc503700013)

[General Application & Categories 2](#_Toc503700014)

[Unstructured/Simple Data 3](#_Toc503700015)

[Structured Data 3](#_Toc503700016)

[Structured Data: Arrays 3](#_Toc503700017)

Part a

# Data Structures Introduction

## What is Data?

Data may be defined as the information that you need in order to do your job, organise yourself, or even carry out your daily activities.

Examples of data we use:

* Details of the products bought on “eBay”.
* Enquiries using the telephone directory.
* Bills for utilities such as electricity or the telephone.
* Our DVD/CD/MP3 collection.
* Checking flight times (arrival/departures) at the airport.
* Checking weather reports on your phone or internet.

## Data types – pascal

Data can be generally grouped into five main categories in their predefined state (i.e. Pascal)

|  |  |
| --- | --- |
| **Data Types (Pre-defined)** | **Description** |
| INTEGER | (Holds integer numbers, eg: -9, 50, 0) |
| REAL | (Holds real numbers, eg: 87.54, 1.566, -0.98) |
| CHAR | (Holds a single ASCII character, eg: 'E', '&', 'z') |
| STRING | (Holds many characters, eg: 'Old Road St') |
| BOOLEAN | (Holds the value TRUE or FALSE) |

**Note:**

A single character takes up 1 byte in memory.

A string is not strictly an unstructured data type. It is really an array of characters, i.e. structured data.

## General Application & Categories

Data type determines the values that can be used with the corresponding type of data and the type of operations that can be performed on such data.

Data can be divided into two main categories based on general application, these are as follows;

* Unstructured (simple) data
* Structured data

To learn more:

<http://datastruct.hnd-computing.info>[/](http://datastruct.hnd-computing.info/)

<https>[://www.tutorialspoint.com/csharp/csharp\_data\_types.htm](https://www.tutorialspoint.com/csharp/csharp_data_types.htm)

Part B

# Unstructured/Simple Data

**Unstructured** **Data** are divided into the following categories:

* **Numeric**
  + Integers
  + Real or Floating Point
* **Character**
  + Single Characters
  + Strings
* **Logical** (Boolean)

**Simple data** types have the following characteristics

* A **name or identifier** by which it is referred to, e.g. Mark
* A **type** which specifies the type of data which can be stored, e.g. Integer
* A value which is assigned to it and if it is a variable may be changed, e.g. 50

# Structured Data

This applies to groups (or collections) of data items that re linked together (i.e. related) for some purpose.

Although simple data types are invaluable in programming there are situations when a programmer can be assisted by the ability to organise and structure data within programs.

**Examples of Structured Data**

* Arrays
* Records
* Tables (Hash Tables)
* Files

## Structured Data: Arrays

An array refers to a collection of variables of similar class (type) and referred to as common name.

An array is a fixed-sized collection of data items, all the **same type**, grouped under the same name.

**Note:**

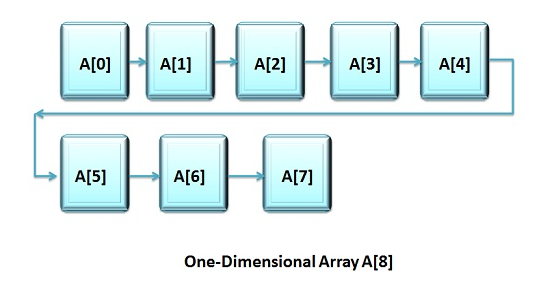
Each individual data item, can be accessed using an index value, called a subscript, which gives the position of the individual item in the structure.

**Examples include:**

* Shopping list
* Lottery numbers

### 1 Dimensional Arrays (1D)

One-Dimensional or Single-Dimensional array is considered as the “list of variables of similar data type”, each variable can be distinctly accessed by specifying its index in square brackets preceded by the name of that array.

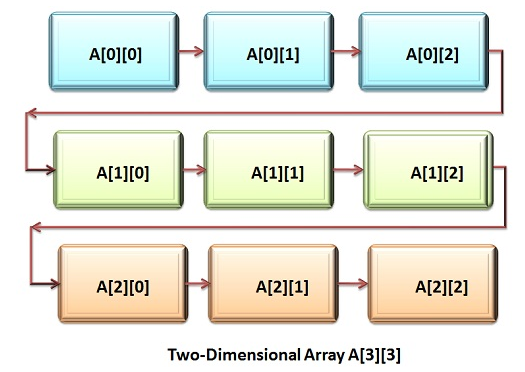


### 2 Dimensional Arrays (2D)

A two-dimensional array can be considered as ‘**array of arrays**’ or ‘**array of one-dimensional arrays**’

A 2D array is stored in the form of the row row-column matrix,

Where the first index indicates the row and the second index indicates the column.



### Array Examples

**Example – 1D Array** (Lottery Array)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Subscript | 1 | 2 | 3 | 4 | 5 | 6 |
| Value | **26** | **36** | **18** | **42** | **45** | **19** |