



# Software Development Teach Yourself Series

## Topic 3: Data and Control Structures Units 1, 3-4

**A:** Level 14, 474 Flinders Street Melbourne VIC 3000  
**T:** 1300 134 518 **W:** [tssm.com.au](http://tssm.com.au) **E:** [info@tssm.com.au](mailto:info@tssm.com.au)

# Contents

Data Structures .....	4
As it appears in Units 1, 3-4.....	4
1D Array.....	4
2D Array.....	4
Records.....	5
Associative Arrays or Dictionaries .....	5
Hash Tables.....	5
Control Structures .....	6
Sequence .....	6
Selection.....	7
Iteration .....	7
Counted Loop.....	7
Pre-Test Loop.....	7
Post-Test Loop .....	8
Review Questions .....	8
Applied Questions .....	9
Solutions to Review Questions .....	10
Applied Question 1 and 2.....	10

## CASE STUDY



All Teach Yourself Series in this package will refer to the following case study.

Tariq Mulner is the manager of a school canteen. He manages how many lunches are going to be prepared each day. It is difficult to tell how many lunches will be sold, so he would like a software solution that students can use to order lunches. This application would provide him with a complete list of orders.

Most of the students have smart phones, so Tariq is suggesting the solution is a phone app that can read in the lunch order, and send it to his device so he can print out the order list.



*Photo by Tirachard Kumtanom from Pexels used with permission*

# Data Structures

## As it appears in Units 1, 3-4

In the Software Development Study Design we focus on the following data Structures:

1D Arrays

2D Arrays

Records

Associative Arrays or Dictionaries

Hash Tables

### 1D Array

An array can only hold one type of data, however it has a built-in index system that makes them very handy for handling data.

Here is an example of a single dimension array: Equipment(6) – it contains 6 items and each item has an index from 0 to 5.

0	1	2	3	4	5
Soccer Balls	Tennis Rackets	Nets	Hockey Sticks	Cricket balls	Cricket Bats

To identify the content using pseudocode:

Equipment(0) ← Soccer Balls

Equipment(1) ← Tennis Rackets

Equipment(2) ← Nets

Equipment(3) ← Hockey Sticks

Equipment(4) ← Cricket balls

Equipment(5) ← Cricket Bats

### 2D Array

A multi-dimensional array can hold tables of data. Like the 1D Array they can only hold one type of data and all items are indexed. All data are strings.

Products (3,4)	0	1	2	3
0	Eggs	\$2.50	500g	17
1	Milk	\$1.60	1l	22
2	Bread	\$2.70	150g	15

You can see how the data has been organized in this 2D array. Each product name is in column 0, the price in column 1 etc.

Product(0,0) = Eggs

Product(2,1) = \$2.70

Product(1,0) = Milk

Product(2,3) = 15

## Records

Records are structures that hold all the data related to an entity or transaction. They can hold many different data types but require unique identifiers

Example

ProductID	ProductName	ProductWeight	ProductPrice
01	Large Eggs	200g	\$2.50
02	Free Range Eggs	200g	\$3.50

## Associative Arrays or Dictionaries

Associative arrays or Dictionaries create structures between data without an index. You can identify whatever index you like, which can be very useful. Unfortunately, if the data type is string, you will not be able to run counters to find or sort data in the dictionary.

Dictionary: Products (string)

Eggs	Bread	Milk	Pies	Chips	Tea
\$3.89	\$2.90	\$1.50	\$3.50	\$2.50	\$3.78

ProductItem.Add (Eggs, \$3.89)

ProductItem.Add (Bread, \$2.90)

ProductItem.Add (Milk, \$1.50)

ProductItem.Add (Pies, \$3.50)

ProductItem.Add (Chips, \$2.50)

ProductItem.Add (Tea, \$3.78)

## Hash Tables

A hash table uses the data to identify a location for storage. These structures are used for storing passwords and usernames. Hashing is the process of translating the data into a value to be used as a locator indicator.

For example:

UserName	Password
JonesL	Blink182
SmithW	Swift989

To store our UserName we are going to hash our Password. To do this we need to convert it to a single value. In the Software Development Study Design we often use the Mod function to limit the size of the arrays we will use to store our data.

If we use a=1, b=2, c=3 etc

Blink182 will be:  $B(2)l((12)i(9)n(14)k(11)l(1)8(8)2(2) = 2+12+9+14+11+1+8+2= 59$ .

Hashing usually uses a prime number – in this case let's use: 5.

$59 \text{ Mod } 5 = 4$

$59 \div 5 = 11 \text{ and } 4 \text{ remainder}$

Now we have a location to place our Usernames:

0	1	2	3	4
			SmithW	JonesL

And Passwords:

0	1	2	3	4
			Swift989	Blink182

The same process is used for hashing Swift989.

## ***Control Structures***

There are three control structures:

1. Sequence
2. Selection
3. Iteration

### ***Sequence***

Sequence refers to the order of instructions. For example:

Read in UserID  
Read In Password

When aspects of a software solution need to be used repeatedly, a subroutine can be created or a function. These can be called up by the main program when required. Correct ordering on when these are called is important to get the right outcomes.

For example:

Read in Product  
Read in Quantity  
On Button Click: Call "CalculateTotalCost"  
Display TotalCost, GST  
On Button Click: Call "Confirm Order"  
Display "Order Confirmed"

## **Selection**

Otherwise called Decision, this is the structure that uses a condition to control the direction of the programming. For example:

```
IF (Not Found= True) Then
    Message ('Your UserName has not been found')
ELSE
    Message ('please enter your password')
END IF
```

## **Iteration**

Iteration makes the most of what computers do best which is repeat instructions. Otherwise called loops, these structures come in three types:

1. Counted
2. Pre-test
3. Post-Test

### **Counted Loop**

This interaction loops a 'known' number of times. The example below will read in a value and store it for when Counter = 0, 1, 2 and 3. (A total of 4 times). A FOR loop does not need an instruction to add one to counter, it does it automatically.

```
FOR (Counter = 0 to 3)
    Read in a value
    Store value in Array
Next Counter
```

### **Pre-Test Loop**

This loop tests a condition before running the loop. The example below will test the variable Total first before reading in a value and adding it to Total. It will repeat the loop until Total is no longer less than 100.

```
While (Total < 100)
    Read in Value
    Total ← Total + Value
End While
```

It's possible that the code inside the loop will never happen, if the condition is never met.

### **Post-Test Loop**

This loop runs at least once and tests at the end. The example below will read in a value and add it to Total before testing it, which could mean Total could be over 100 before the condition is tested.

Repeat

    Read in Value

    Total  $\leftarrow$  Total + Value

Until Total >100

The code inside the loop will happen at least once, depending on when the condition is not met.

### **Review Questions**

1. Which of the following is a 1D array?

- A. Products(12,4)
- B. Products(Eggs, \$2.50)
- C. Products(5)
- D. Product.Eggs

2. Which of the following is a 2D array?

- A. Products(12,4)
- B. Products(Eggs, \$2.50)
- C. Products(5)
- D. Product.Eggs

3. Which of the following control structures is a loop?

- A. Iteration
- B. Array
- C. Decision
- D. Sequence



4. Which of the following control structures uses an IF Statement?
- A. Iteration
  - B. Array
  - C. Decision
  - D. Sequence

### **Applied Questions**



Tariq has asked you to produce an app. The Analysis is available in TOPIC 1. You have 3 weeks to develop the app to hand over. Ideally Tariq should have the solution for 2 weeks before evaluation begins.

5. Describe and illustrate an array that could be used by Tariq’s application.


6. How could the Canteen App use a Dictionary?


7. What would be the best use of a Hash Table for Tariq’s Canteen App?




## ***Solutions to Review Questions***

**1. C. Products(5)**

**2. A. Products(12,4)**

**3. A. Iteration**

**4. C. Decision**

## ***Applied Question 1 and 2***

**5.** There are many opportunities for use of an array in this case study question:

1D Array for Student Numbers – so they can be checked if the student has entered the data correctly

2D Array to store the orders so that a running total of each item can be calculated.

Associative Array to store product item information

**6.** A dictionary is an associative array – ideally it could be used to store product information (Item name, price)

**7.** Hash Tables are often used for user names and passwords. Students would need to have passwords to use the system, so a hash table could be used to store the student IDs and passwords.