**Lesson Summary and Revision Notes: Week 14**

3 Common Methods of Writing Algorithms

* **Structured English**: a method of showing the logical steps in an algorithm, using an agreed set of English words for commands and mathematical operations to represent the solution.

* **Flowchart**: shows diagrammatically, using a set of symbols linked together with flow lines, the steps required for a task and the order in which they are to be performed.

* **Pseudocode**: a method of showing the detailed logical steps in an algorithm, using keywords, identifiers with meaningful names and mathematical operators to represent a solution.

**Flow Chart to Compare Two Numbers and Print Out the Larger Number**



**Writing Algorithms Using Pseudo Codes**

Selection using IF statements (single choice):

IF MyValue > YourValue THEN

OUTPUT "I win"

ENDIF

Selection using IF statements (single choice with alternative):

IF MyValue > YourValue THEN

OUTPUT "I win"

ELSE

OUTPUT "You win"

ENDIF

Selection using CASE statements (multiple choices):

CASE OF Direction

"N": Y ← Y + 1

"S": Y ← Y – 1

"E": X ← X + 1

"W": X ← X – 1

ENDCASE

Selection using CASE statements (multiple choices with alternative):

CASE OF Direction

"N": Y ← Y + 1

"S": Y ← Y – 1

"E": X ← X + 1

"W": X ← X – 1

OTHERWISE : OUTPUT "Error"

ENDCASE

Iteration (Loop) Using FOR Keyword

Total ← 0

FOR Counter ← 1 TO 10

OUTPUT "Enter a number "

INPUT Number

Total ← Total + Number

NEXT Counter

OUTPUT "The total is ", Total

Iteration (Loop) Using FOR Keyword with STEP Increment:

FOR Counter ← 1 TO 10 STEP 2

OUTPUT Counter

NEXT Counter

**Note**: The value of Counter will be: 1, 3, 5, 7, 9.

Iteration (Loop) Using REPEAT-UNTIL Keyword:

REPEAT

OUTPUT "Please enter a positive number "

INPUT Number

UNTIL Number > 0

**Note**: Statements in a REPEAT loop are always executed at least once

Iteration (Loop) Using WHILE Keyword:

Number ← 0

WHILE Number >= 0 DO

OUTPUT "Please enter a negative number:”

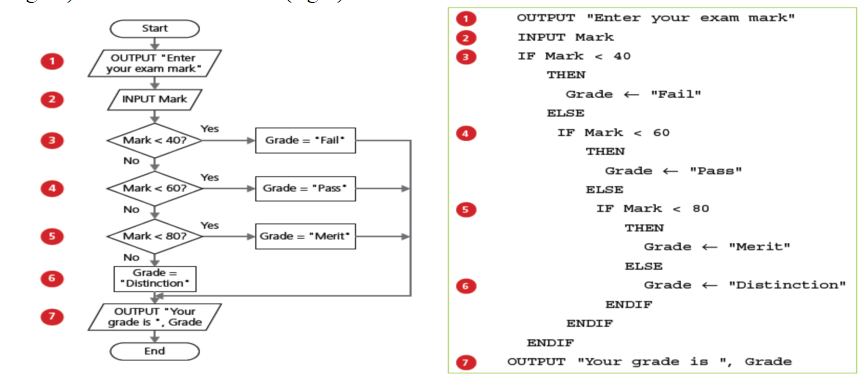
INPUT Number

ENDWHILE

**Note**: Statements in a WHILE loop may sometimes not be executed.

**Writing Pseudo Code from Flow-Chart**

Students should learn how to write Pseudo Code from Flow-Chart



**6 Common Data Types used in Programming Languages**

|  |  |  |  |
| --- | --- | --- | --- |
| **Data Type** | **Description** | **Pseudocode** | **Example** |
| Boolean | Logical Values; True (1) or False (0) | BOOLEAN | TRUE / FALSE |
| char | Single alphanumeric character | CHAR | a |
| date | Value to represent a date | DATE | 26-06-1990 |
| integer | Whole number, positive or negative | INTEGER | 12 |
| real | Positive or negative number with a decimal point | REAL | 3.7 |
| string | Sequence of alphanumeric characters | STRING | Happy Birthday |

**How to Declare Data Types**

In pseudocode a declaration statement takes this form:

DECLARE <identifier> : <data type>

For example:

DECLARE myBirthday : DATE

DECLARE myAGE : INTEGER

DECLARE myNAME : STRING

**Record Data Types**

* Records are composite data types formed by the inclusion of several related items that may be of different data types.
* A record will contain a fixed number of items.
* For example, a record for a book could include title, author, publisher, number of pages, and whether it is fiction or non-fiction.

**Create Record Data Type Using Pseudo Code**

The following create record structure for a book:

BookRecord = RECORD

DECLARE title : STRING

DECLARE author : STRING

DECLARE publisher : STRING

DECLARE number\_of\_pages : INTEGER

ENDRECORD

**Assess and Set the Items of a Record Data Type Using Pseudo Code**

The following will set the values of a Book record data type

BookRecord.title ← "Invisble Man"

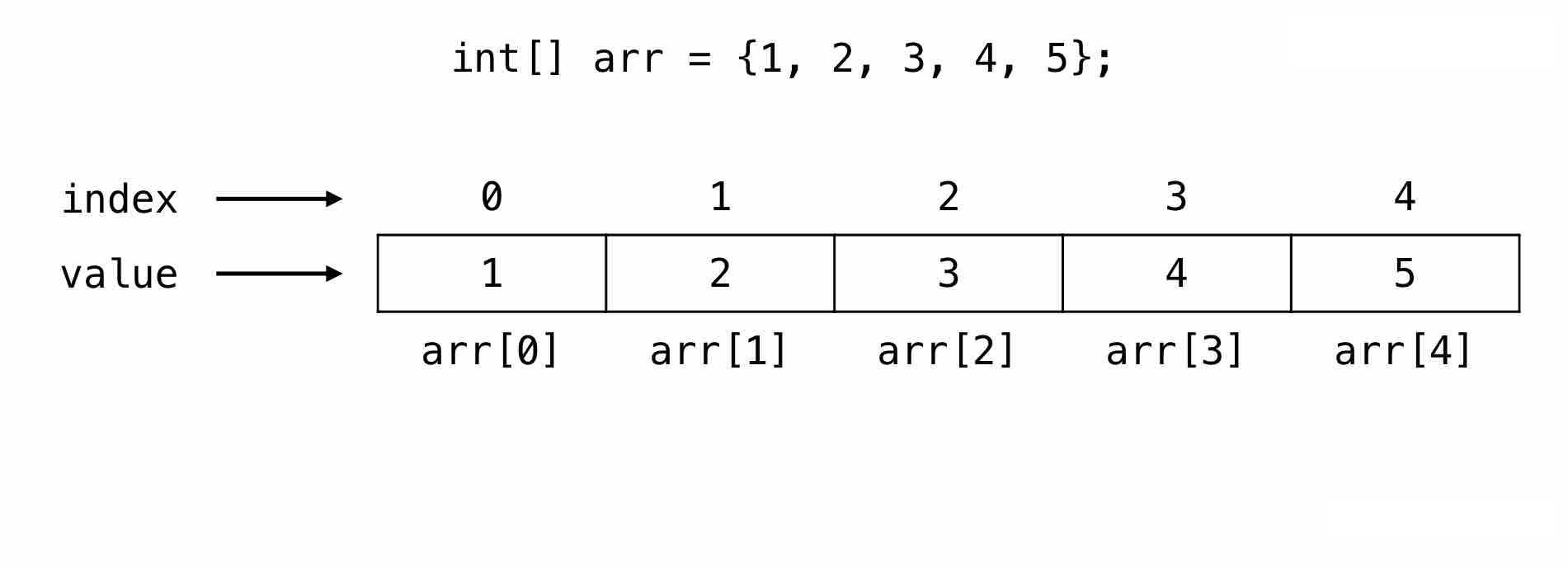
BookRecord..author ← "HG Wells"

**Arrays**

* An array is a data structure containing several elements of the same data type; these elements can be accessed using the same identifier name.
* The position of each element in an array is identified using the array’s index.
* The index of the first element in an array is the lower bound and the index of the last element is the upper bound.
* The lower bound of an array is usually set as zero or one.
* Arrays can be one-dimensional or multi-dimensional.

**1 Dimensional (1D) Array**

* A 1D array can be referred to as a list.
* An example of a list with five elements and a lower bound of zero.
* An array uses an index to point to the element in the array.
* The first element always has an index value of zero.
* Value of first element of the array, arr[0] = 1, and value of last element of the array, arr[4] = 5



**Example: Storing Data into a 1D Array**

Pseudo Code to declare an array and store a list of 10 words into an array.

ARRAY spelling\_words[10] // Declare an array of 10 elements

spelling\_words[0] = "path" // Assign a value into a slot of the array

spelling\_words[1] = "floor"

spelling\_words[2] = "sugar"

spelling\_words[3] = "because"

spelling\_words[4] = "beautiful"

spelling\_words[5] = "clothes"

spelling\_words[6] = "whole"

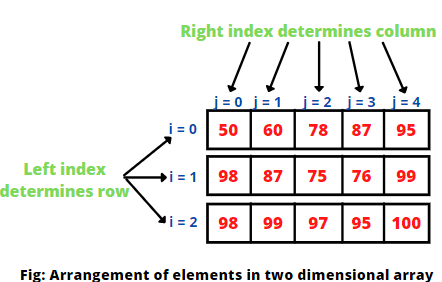
spelling\_words[7] = "behind"

spelling\_words[8] = "move"

spelling\_words[9] = "busy"

**2 Dimensional (2D) Array**

The figure below where the arrangement of elements in a 2D array.



* As you can observe that rows are starting from 0 to 2 and columns are starting from 0 to 4.
* Any element can be referred to as array[i][j] where i and j are two indexes that represent row position and column position respectively.
* In a two-dimensional array, we access an element through a row and column index.
* **Example**: array[0, 2] = 78 array[1,3] = 76 array[2,3] = 95

**File I/O (Input and Output)**

When we open a file, there are three types of file access modes listed below:

* **Read**: read data from the file
* **Write**:writes data to the file, any existing data stored in the file will be overwritten
* **Append**: adds data to the end of the file