DATA STORAGE IN PROGRAMMING

Data used in by programs need to be held (stored) in the computer's memory(RAM) while the program is running. In programs this is as a variable or constant.

Variables

- A variable is a name given to a data storage location. It can be set up (**declared**) to hold different types of data. Each data structure is allocated a different amount of memory. Variables can set up to hold either **Simple or Structured data** types.
- Variable names can be up to 255-ish characters long (depending on program) and they should be meaningful to their context in the algorithm although they need not be legitimate English words.

Constants

- A constant - a data item that retains the same value throughout the program for calculations, e.g. Pi(π) = 3.14, Interest Rate, Tax Rates

Question: Outline some advantages of using constants in a program.

____

____

Data dictionaries are used to describe the variables being used in an algorithm.

Variable Name	**Data type**	**Size**	**Description**
Age	Integer	4 bytes	Age of the competitor e.g. 16

DATA STRUCTURES

SIMPLE DATA TYPES

Simple data can only store one piece of data, e.g. a name, or an age. Simple data types include:

Data types	Description	Typical memory requirements	Examples
Character	A character represents a single alphanumeric (a - z, 0 - 9), plus some special characters.	8 bits	Y w 9
String	A string represents a set of characters. The number of characters in the string can be set. E.g. The string length for a surname could be set to a maximum length of 20 characters.	8 bits times the length of string	Joshua ID4322f 122-23bc
Integer	An integer represents only whole numbers, both positive and negative.	16 or 32 bits	3 -55 2000
Real	A real, also called a floating point, represents any positive and negative number, including both integer and decimal numbers.	32 or 64 bits	8 3.44443 -332.44
Boolean	A Boolean, also called logical, represents one of 2 possible values, 0 or 1 to represent true or false.	1 bit	0, 1

Activity 1: For each of the following data items, write down the data type that would be best suited to hold data. Justify your answer.

Data	Data type	Reason
A counter variable		
No. of students at Bradfield		
student gender		
name of car model		
whether fees have been paid for 2003		
cost of bread		
Population of the world		
Date of birth		
Average rainfall for a month		

STRUCTURED DATA TYPES

These are more complex data storage types. They can hold more than one item of data and are used when you need to hold a number of related sets of data. The structured data types we will be studying are:

- Arrays
  -One and two dimensional
  -Array of records
- Records
- Files (random and sequential)

ARRAYS

Arrays are used to hold a number of related data items, which are held as a list (one dimension) or table/grid (2 dimensions). Arrays can only hold one data type. Each data item in the array is called an element. It is accessed by the **variable (array) name** and a location in the array called an index. E.g. Team(3)

Most programming languages start the index at 0. However arrays can be declared to start at 1.

Example: The following is an example of an array called Team that has 9 elements with the index starting at 1

Team	1	2	3	4	5	6	7	8	9
	Fred	Mary	Bill	Phil	Jim	Phip	Jack	Jill	Bob

Activity 2:

1. For the Team array above

PRINT Team (2)   PRINT Team (5)
would print ___________   this would ___________

2. The index in the brackets can be either a number or a variable name (which holds a number).

e.g..   index = 3
PRINT Team (index)
this would print out _____________

3. You can change the contents of an array. e.g. Change Phil in Team to Elmer.

index = 4
Team (index) = Elmer
So Team array is now

1	2	3	4	5	6	7	8	9
Fred	Mary	Bill	Elmer	IT	Phip	Jack	Jill	Bob

A **two dimensional array** is like a table with each storage location having an x and y co-ordinate.

e.g. the array called Grid

	1	2	3	4
1	A	B	C	
2				F
3	F		Y	O
4	P			K
5		U		R

The data value C is stored in Grid(1,3)

Activity 3

```
// The program...
DECLARE ARRAY Team(1 TO 9) AS STRING * 10
count = 0
// The following loop enters the names of players into the array
WHILE ( count < 9 )
        count = count + 1
        PRINT "Please enter the name of player number "; count
        INPUT Team(count)
WEND
count = 0
// The following loop prints the names of players in the array
WHILE count < 9
        count = count + 1
        PRINT "Player number "; count; " is "; Team(count)
WEND
```

1. Describe what the above program does. (Problem description)

____

____

____

____

2. Complete the following Data Dictionary for the variables used

Name	Data type	Size	Description

3. Create a flowchart of the above code.

4. Outline the advantage of using an array in this question.

____

____

____

____

5. We will now add the following code to allow the array to be searched.

```
INPUT “Enter name of player to find”; find
Index = 0
WHILE(index < 9)
        Index = index + 1
        IF find = Team(index) THEN
                PRINT “Found”
        END IF
WEND
```

6. Outline what happens if the name entered is not in the array.

____

____

____

____

7. Modify the pseudocode above to enter the **ages** of each Team player AND to enter 15 people.

8. For each of the following sets of data
   - Write the simple data type being held
   - Write the pseudo code to declare an array to hold this set of data

   A. Population:

400 000 000	1 500 000 000	19 000 000	20 000 000

____

____

____

____

B. Results: 31.5, 70, 80, 65, 0, 95.5

____

____

____

____

RECORDS

To hold related data that has different data types a **record** can be used. Each data item in a record is called a field and is accessed by the record and the field name (e.g. Detail[index].name) An example of a record is:

Details

Name	**Name**	**Age**	**Address**
Smith	John	16	1 Short St

To DECLARE THIS RECORD

```
TYPE details
        Name AS STRING * 20
        Age AS INTEGER
        Address AS STRING * 40
END TYPE
```

To ACCESS THIS RECORD

E.g. Print Details.Age   would print 16

NOTE:

- A record can only hold one **set** of details.
- To hold a **number** of people's details an **Array of Records** needs to be created.

ARRAY OF RECORDS

In an array of records each element of the array is set up to hold the same record type. This is similar to how a database is set up to hold data.

e.g.. The following is an example of an array of records, the array name is called Baseball.

	Name	Age	Address
1	Jill	12	1 Short St
2	Jack	14	5 Ade Ave
3	Bill	8	3 Ed St
4	Bob	17	9 Sit Ave
5	…	…	…

E.g. Baseball(1) refers to record which holds Jill, 12 and 1 Short St.

To access an item in an array of records you use the array name with the index location, followed by a dot and then the field name.

```
E.g.  PRINT Baseball(1).Age
          Would print out _____________
          or
Index = 4
PRINT Baseball(index).name
          Would print out _____________
```

Activity 4: For the Job_Class data, determine:

- the simple data type is being held

____

____

- write the pseudocode declaration statement for the array of records

Job_Class

Name	Position	Salary
Dav	P/T	25 000
Jone	F/T	30 000
Jack	P/T	6 000
Jo	F/T	70 000
Jaine	F/T	80 000

FILES

Simple data types, arrays and records all set up temporary storage space in primary memory, so when the machine is turned off, or the program is closed the data is lost.

Files store data in secondary memory (e.g. disk drives), so data is stored on a permanent base. Files are used when you wish to store data for later use, or when there is not enough primary memory to store all data needed for a program.

A file is usually a collection of data subdivided into records (and then fields).

There are two main types if files we will look at **random** and **sequential** files. File sizes are more flexible and can be updated and changed depending upon the data being held. You do not have to define the number of records in a file, whereas the number of elements in a record is a set at the start of the program.

When using data from a file, it must first be loaded into RAM (one record at a time).

We will look in Files, both sequential and random later in the course.