ENCE 202
FORTRAN
Handout 3

Introduction to FORTRAN

by

Dr. Ibrahim A. Assakkaf

ENCE 202
Spring 2000
Department of Civil and Environmental Engineering
University of Maryland

ENCE 202
FORTRAN
Handout 3

Nonnumeric Data Types

- Logical Data Type
  - Logical variables can take one of the following two values
    - .TRUE.
    - .FALSE.
  - Declare logical variables using:

LOGICAL *list-of-variables*

ENCE 202
FORTRAN
Handout 3

Nonnumeric Data Type

- Logical Data Type
  - Assignment Statements
    - logical-variable = logical-expression
    - e.g.

      X = .TRUE.

      Y = x
  - Logical Operators
    - .NOT. .AND. .OR. .EQV. .NEQV.

ENCE 202
FORTRAN
Handout 3

Nonnumeric Data Type

• A. J. Clark School of Engineering • Department of Civil and Environmental Engineering

- Logical Data Type
  - Examples
    - Z = X .OR. Y
    - Z = (X .LT. 1.5) .AND. .NOT. Y

  The latter statement means assign the value true to Z if an only if X < 1.5 and Y is false. Note X = real variable, Y = logical variable

ENCE 202
FORTRAN
Handout 3

Character Data Type

- Declare using the type statement

  CHARACTER *n*, *list*

  where *n* = length of each variable in the list

  – Examples:
    - X = `COMPUTER`
    - X (4:6) = `PUT`
      – They are treated like arrays
    - X (:4) = `COMP` (default is 1 for the first #)
    - X (n:n+2) = `COM` (for n = 1)

ENCE 202
FORTRAN
Handout 3

Character Data Type

– We can compare character values using .LT., .GT., .EQ., .LE., and .NE. They are based on the sequence of ASCEII number equivalent.

– Printing Logical Values

```
PRINT *, A, B
```

ENCE 202
FORTRAN
Handout 3

One Dimensional Arrays

- Variables can be classified as:
  - Simple Variables
  - data structure or arrays
- To establish a data structure use

  DIMENSION X(1:10)

  REAL X

  OR

  REAL X(10)

  DIMENSION X(10)

ENCE 202
FORTRAN
Handout 3

One Dimensional Arrays

• A. J. Clark School of Engineering • Department of Civil and Environmental Engineering

which means that X can take values X(1), X(2), ... X(10). In general, the statement is as follows:

DIMENSION List

The list consists of array-name (l : u), where l = lower size, and u = upper size as integers. l and u are integers, e.g., -2, -1, 3, 10, 1000, 0.

ENCE 202
FORTRAN
Handout 3

One dimensional Arrays

- Input and Output of Arrays
  - Three Methods:
    1. Do loops
    2. Array names
    3. Implied Do loops

ENCE 202
FORTRAN
Handout 3

One Dimensional Arrays

- Examples: Input and Output of Arrays
  - Example 1

```
      DIMENSION X(5)
      DO 10 I = 1, 5
10    READ *, X (I)
```

ENCE 202
FORTRAN
Handout 3

One Dimensional Arrays

- Examples: Output and Input of Arrays
  - Example 2

```
READ *, X(1), X(2), X(3), X(4), X(5)
```

  - Example 3

```
      READ 10, X
10    FORMAT (5F6.1)
```

Note: It reads $X(1)$ to $X(5)$

ENCE 202
FORTRAN
Handout 3

One Dimensional Arrays

• A. J. Clark School of Engineering • Department of Civil and Environmental Engineering

- Examples: Output and Input of Arrays
  - Example 4

```
      READ *, (X(I), I = 1, 5)
```

or

```
      N = 5
      READ *, (X(I), I = 1, N)
```

ENCE 202
FORTRAN
Handout 3

One Dimensional Arrays

- In general the READ or PRINT can be:

READ *, (input/output list, control-variable = initial value, limit)

PRINT *, (input/output list, control-variable = initial value, limit)

Example

READ (*, *, END = 10) (X(I), I = 1, 100)

The END means that once data are read go to statement number 10

ENCE 202
FORTRAN
Handout 3

One Dimensional Arrays

• A. J. Clark School of Engineering • Department of Civil and Environmental Engineering

- PARAMETER and DATA Statements

These statements are used to input constants in programs. The values can be real, integer, or character

PARAMETER (par1 = val1, par2 = val2, …)

ENCE 202
FORTRAN
Handout 3

One Dimensional Arrays

- Examples: PARAMETER Statements

Example

```
REAL PI
PARAMETER (PS = 3.14)
```

Example

```
CHARACTER * (*) X
PARAMETER (X = 'TEST')
```

The (*) in the firs statement means that length is the same as the value of X. In the second statement the length is 4

ENCE 202
FORTRAN
Handout 3

One Dimensional Arrays

- DATA Statement

DATA $list_1/data_1/$, $list_2/data_2$, ..., $list_k/data_k/$

ENCE 202
FORTRAN
Handout 3

One Dimensional Arrays

- Examples: DATA Statements

Example

```
REAL  X(5)
DATA  X/1,2,3,4,5/
```

Example

```
INTEGER N, I
REAL  X(10)
DATA  N, (X(I), I = 1, 5)/5, 5 *0.0/
```


ENCE 202
FORTRAN
Handout 3

Multi-Dimensional Arrays

- Multi-dimensional arrays can be declared using DIMENSION statements or type (e.g., REAL, INTEGER, etc.) statements.
- Example

```
REAL array-name (l1 : u1, l2 : u2, ..., lk : uk)
```

ENCE 202
FORTRAN
Handout 3

Multi-Dimensional Arrays

- Example

REAL X(2,5)

The result of statement is declaring X as a 2 by 5 array. Also, it can declared using one of the following statements:

DIMENSION X(2, 5)

REAL X(1:2, 1:5)

ENCE 202
FORTRAN
Handout 3

Multi-Dimensional Arrays

• A. J. Clark School of Engineering • Department of Civil and Environmental Engineering

- Processing multi-dimensional arrays
  - Two-dimensional arrays can be processed either (1) row-wise, (2) column-wise. Column-wise processing is the default.
  - Multi-Dimensional arrays, process them column wise that is first subscript is varied first, then second subscript followed by third ... and so on.

ENCE 202
FORTRAN
Handout 3

Multi-Dimensional Arrays

- Input and Output of Arrays
  - The input and output of arrays can be performed using (1) DO loops, (2) array name, or (3) Implied Do loops
  - Example

```
      REAL X(4,3)
      DO 20 I = 1,4
      DO 10 j = 1, 3
      READ *, X(I, J)
10    CONTINUE
20    CONTINUE
```


ENCE 202
FORTRAN
Handout 3

Multi-Dimensional Arrays

- Example

```
INTEGER X(2,3)
  READ *, X
```

The above statements results in reading X column wise. If we read 3, 4, 8, 9, 10, 5, the result is the following matrix:

$$\Rightarrow \begin{bmatrix} 3 & 8 & 10 \\ 4 & 9 & 5 \end{bmatrix}$$

Multi-Dimensional Arrays

- Example
  - In general use a read or write statement with (input/output list, control-variable = initial-value, limit) as an implied DO loop. For example

$$\text{READ*, ((X(I, J), J} = 1,4), \text{I} = 1,3)$$

$\uparrow_{\text{column}}$ $\uparrow_{\text{row}}$

Multi-Dimensional Arrays

This is the same as

READ *, (X(I,1), X(I,2), X(I,3), X(I,4)), I = 1, 3)

which is the same as

READ *, X(1,1), X(1,2), X(1,3), X(1,4), X(2,1), X(2,2),
+X(2,3), X(2,4), X(3,1), X(3,2), X(3,3), X(3,4)