Basic Fortran 77

T2KSK @ SB Meeting

José Palomino
04/15/2013
Fortran was the first ever high-level programming language.

Originally developed by IBM at their campus in south San Jose, California in the 1950s for scientific and engineering applications.

High level language instructions are not executable. Instead, a high level language source program is read as input by a program called a compiler.
Variables, types, and declarations

Variable names:

Fortran 77 does not distinguish between upper and lower case, in fact, it assumes all input is upper case.

Spaces are ignored by Fortran 77, e.g. COL UMN is equivalent to COLUMN.

Reserved words: "program", "real", "stop" and "end".
Variables, types, and declarations

Types:

Integer, real, double precision, complex, logical and character.

When included in a Fortran statement, a string must be delimited by single quotes ('), A single quote may be included in a string by writing two consecutively (").

An INTEGER value can be assigned to a REAL variable ( \( X = 4 \) equivalent \( X = 4.0 \) ).

A REAL value can be assigned to an INTEGER variable ( \( A = 0.99 \) equivalent \( A = 0 \) ).
Logical expressions

Logical expressions can only have the value .TRUE. or .FALSE.

\[
\begin{align*}
.LT. &\quad \text{meaning} &\quad <.\leq &\quad <=.GT.> \\
.GE. &\quad \geq &\quad = &\quad /= \\
.EQ. &\quad = &\quad &= \\
.NE. &\quad /= &\quad &
\end{align*}
\]

The rule is that arithmetic expressions are evaluated first, then relational operators, and finally logical operators.

```fortran
logical a, b  
a = .TRUE.  
b = a .AND. 3 .LT. 5/2
```

logical operators .AND. .OR. .NOT.
Statements

The Parameter statement

Some constants appear many times in a program. It is then often desirable to define them only once, in the beginning of the program.

```
program circle
real r, area, pi
parameter (pi = 3.14159)

! This program reads a real number r and prints
! the area of a circle with radius r.
```

Exist: **print**, **program**, **end** and **stop** statements.
IF

IF statement

IF (logical_expression) executable_statement

example: IF (A.LT.B) SUM = SUM + A

IF structure

IF (Lo) THEN
  So
ELSE IF (Li) THEN
  Si
ELSE
  Sn
END IF

example: IF (A.LT.B) THEN
  SUM = SUM + A
  PRINT *, SUM
END IF
The numerical value of statement labels have no significance, so any integers can be used, in any order. Typically, most programmers use consecutive multiples of 10.
GOTO

\textit{label if (logical expr) then statements goto label endif}

\textbf{example:}

\begin{verbatim}
  n = 1
  10 if (n .le. 100) then
    write (*,*), n
    n = 2*n
  goto 10
  endif
\end{verbatim}
Other ways....

do while (logical expr)
  statements
enddo

while (logical expr) do
  statements
enddo

do
  statements
until (logical expr)
We can declare arrays by two ways:

1: \texttt{real a(20)}

2: \texttt{real A, x}
   \texttt{dimension x(50)}
   \texttt{dimension A(10,20)}

Fortran arrays are indexed from 1 and up. However, you may define an arbitrary index range for your arrays using the following syntax:

\texttt{real b(0:19), weird(-162:237)}

2 dimensions:

\texttt{name (low\_index1 : hi\_index1, low\_index2 : hi\_index2)}

Fortran 77 allows arrays of up to seven dimensions!!!